

Université
de Toulouse

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Présentée et soutenue le *30 septembre 2015* par :

Nikola TULECHKI

**Natural language processing of incident and accident reports:
application to risk management in civil aviation**

*Traitement automatique de rapports d'incidents et accidents: application à la
gestion du risque dans l'aviation civile*

JURY

Patrice BELLOT	PR, <i>LSIS</i> , Marseille	Rapporteur
Yannick TOUSSAINT	CR HDR, <i>INRIA</i> , Nancy	Rapporteur
Cécile FABRE	PR, <i>CLLE-ERSS</i> , Toulouse	Examinatrice
Ludovic TANGUY	MCF HDR, <i>CLLE-ERSS</i> , Toulouse	Directeur
Eric HERMANN	Directeur de <i>CFH-SD</i> , Toulouse	Invité

École doctorale et spécialité :

CLESCO : Sciences du langage

Unité de Recherche :

CLLE-ERSS (UMR 5263)

Directeur de Thèse :

Ludovic TANGUY

Rapporteurs :

Patrice BELLOT et Yannick TOUSSAINT

Abstract

This thesis describes the applications of natural language processing (NLP) to industrial risk management. We focus on the domain of civil aviation, where incident reporting and accident investigations produce vast amounts of information, mostly in the form of textual accounts of abnormal events, and where efficient access to the information contained in the reports is required.

We start by drawing a panorama of the different types of data produced in this particular domain. We analyse the documents themselves, how they are stored and organised as well as how they are used within the community. We show that the current storage and organisation paradigms are not well adapted to the data analysis requirements, and we identify the problematic areas, for which NLP technologies are part of the solution.

Specifically addressing the needs of aviation safety professionals, two initial solutions are implemented: automatic classification for assisting in the coding of reports within existing taxonomies and a system based on textual similarity for exploring collections of reports.

Based on the observation of real-world tool usage and on user feedback, we propose different methods and approaches for processing incident and accident reports and comprehensively discuss how NLP can be applied within the safety information processing framework of a high-risk sector. By deploying and evaluating certain approaches, we show how elusive aspects related to the variability and multidimensionality of language can be addressed in a practical manner and we propose bottom-up methods for managing the overabundance of textual feedback data.

Acknowledgements

Much like rock climbing, writing a thesis is both a solitary endeavour and a team exercise. Making progress up the mountain is not possible without someone standing on *terra firma* and holding the other end of the rope. Completing a thesis is not possible without a strong, rich and varied supportive context. For that, before we begin, I would like to express my gratitude to all the people who have helped me in doing this research.

First and foremost I would like to thank Ludovic Tanguy who has been my adviser not only for this thesis but for my entire life in academia. Starting from the very first introductions to the domain of natural language processing and to computer programming in 2007 right up to this very moment, Ludovic's advice, guidance and support have been invaluable. Whether presented with an outrageous "great" idea or with a last minute crisis, Ludovic has always had a trick up his sleeve and with the fitting words has pointed me in the right direction. For all that and much more, thank you, Ludovic!

I would like to thank the members of the jury, Patrice Bellot and Yannick Toussaint for accepting to write the detailed reports and Cécile Fabre for accepting to be my examiner.

This thesis would not have been possible without the material support and the context provided by both *CFH/Safety Data* and the *CLLE-ERSS* linguistics research laboratory.

From *CFH/Safety Data*, I would like to thank Eric Hermann and Michel Mazeau for providing me with this opportunity and for believing in the potential of natural language processing in the context of risk management. I would also like to express my gratitude to them for introducing me to the fields of industrial ergonomics and human factors and thus providing the foundations for my understanding of the complexities of human work and interaction with technology. My respect also goes to the rest of the *CFH/Safety Data* team: Céline Raynal, Christophe Pimm, Vanessa Andréani, Marion Laignelet and Pamela Maury for constituting of this exceptionally rich working environment. From *CLLE-ERSS* I would like to thank all the past and present members, whose various inputs throughout both the time of this writing and my academic upbringing constitutes the foundations of this research. A wonderful and colourful crowd, of which I feel honoured being a part.

I would especially like to thank Assaf Urieli and Nicolas Ribeiro without whose help and contributions whole sections of this thesis would not exist as well as my good friend and colleague Aleksandar Kalev for his years long professional and personal support.

My gratitude also goes to Marie-Paule Péry-Woodley for providing a much needed external perspective and helping me overcome the dreaded writer's block and Mai Ho-Dac for always finding a way to show me the bright side of academia.

For their invaluable input in helping me understand the intricacies of aviation and flying I would like to thank Grégory Caudy, Jérôme Rodriguez and especially Reinhard Menzel for patiently sharing his profound knowledge of aviation safety management.

For accepting subjects related to my work for their Masters projects and for their exemplary work, I thank Céline Barès, Joao Pedro Campello Rodriguez and Clement Thibert.

For their support and encouragements, I thank my fellow doctoral students Fanny Lalleman and François Morlane-Hondère and Caroline Atallah.

Finally to all those, whose love has made me wake up with a smile in the morning and to all those whose words have made me fall asleep with a peaceful mind. Thank you!

Contents

List of Figures	11
List of Tables	13
Introduction	17
1 Basics of accident modelling and risk management	25
1.1 What is an accident?	26
1.1.1 From normality to disaster	26
1.1.2 A complicated definition	29
1.1.3 Severity, frequency and visibility	30
1.1.4 The basics of incident reporting	32
1.2 Risk management in a complex systems	34
1.2.1 The descending flow: controlling the processes	36
1.2.2 The ascending flow: information driven decision making	37
1.3 Looking for patterns	38
1.4 Chapter conclusion	40
2 Safety information in civil aviation: actors, models and data	41
2.1 Producing occurrence data	43
2.1.1 The actors	44
2.1.2 Official accident investigations	46
2.1.2.1 The process	46
2.1.2.2 Report examples	46
2.1.2.3 Acquiring the data	53
2.1.3 Preliminary reports and accidents briefs	54
2.1.3.1 The process	54
2.1.3.2 Report examples	54
2.1.3.3 Acquiring the data	56
2.1.4 Voluntary reporting programs	56
2.1.4.1 The process	56
2.1.4.2 Report examples	57
2.1.4.3 Acquiring the data	59

2.1.5	Safety management systems and mandatory reporting	59
2.1.5.1	The process	59
2.1.5.2	Report examples	60
2.1.5.3	Acquiring the data	61
2.1.6	Other sources of occurrence data	61
2.1.6.1	Specialised data providers	61
2.1.6.2	Acquiring the data	62
2.1.6.3	Press	62
2.1.6.4	Community efforts and user generated content	63
2.1.6.5	Acquiring the data	64
2.1.7	A typology of occurrence reports	64
2.1.7.1	External categorisation	64
2.1.7.2	Internal categorisation	66
2.2	Storing and organising occurrence data	70
2.2.1	The occurrence and its lifecycle	70
2.2.2	Accident models, coded data and taxonomies	70
2.2.3	Examples of metadata	72
2.2.3.1	Simple factual information	72
2.2.3.2	Standard descriptors of the accident sequence	72
2.2.3.3	The ASRS coding schema	75
2.2.3.4	SMS systems and the bow-tie model	75
2.2.3.5	ECCAIRS and ADREP	76
2.2.4	A typology of taxonomies	82
2.3	Using occurrence data	84
2.3.1	Querying the collection	84
2.3.2	KPIs and statistics	85
2.3.3	Intelligence and monitoring	89
2.4	Issues when dealing with large collections of occurrence data	90
2.4.1	Issues with natural language reports	91
2.4.2	Issues with coded data and taxonomies	92
2.4.2.1	Complex codification schemes	92
2.4.2.2	Dynamic systems and static taxonomies	92
2.4.2.3	Changing models and taxonomies	93
2.4.2.4	Bottleneck effects	93
2.5	Summary of the issues and NLP as a solution	94
2.6	Chapter conclusion	97
3	NLP: domains of application	99
3.1	Information retrieval	100
3.1.1	Problem definition	100
3.1.1.1	Information need and query formulation	101
3.1.1.2	Models of IR for document processing	102
3.1.1.3	Displaying the results	102
3.1.1.4	IR performance	103

3.1.1.5	An example of full text search problem	103
3.1.2	Linguistic issues in IR	105
3.1.2.1	Morphological variation	106
3.1.2.2	Lexical variation	107
3.1.2.3	Compositionality and semantic variation . . .	109
3.1.2.4	Discourse and document structure	109
3.1.3	IR for occurrence data	110
3.1.3.1	Precise information needs	110
3.1.3.2	Undefined information needs	110
3.1.3.3	Favouring recall	110
3.1.4	A broader perspective on the IR problem definition . . .	111
3.2	Automatic text categorisation	112
3.2.1	Problem definition	112
3.2.1.1	The nature of the classification problem	113
3.2.1.2	The choice of classifier	113
3.2.1.3	Document representation	114
3.2.2	Specifics of applying TC to occurrence data	114
3.2.3	Automatic classification of occurrence categories: an ex- ample	115
3.2.3.1	Context	115
3.2.3.2	Corpus size and category distribution	116
3.2.3.3	Classifier and classification problem	116
3.2.3.4	Results	117
3.2.3.5	Industrialisation	118
3.3	Chapter conclusion	118
4	From text to vectors	119
4.1	Extracting features	120
4.1.1	Tokenising	121
4.1.2	Levels of normalisation	121
4.1.3	Overview of a processing chain	123
4.1.4	Basic processing	124
4.1.5	Word n-gram extractor	126
4.1.6	Detector of developed acronyms	126
4.2	Representing documents in vector space	127
4.2.1	The term matrix	128
4.2.2	Feature weighing	128
4.3	Dimensionality reduction methods	130
4.3.1	Smoothing the term matrix	130
4.3.2	Explicit methods	131
4.3.3	Intrinsic or extrinsic, hidden or explicit?	132
4.4	Chapter conclusion	133
5	The <i>timePlot</i> system: detecting similar reports over time	135

5.1	Calculating similarity	137
5.1.1	At the prototype stage	137
5.1.2	From the prototype to a functioning system	138
5.2	Presentation of the tool	138
5.3	Chronological distributions of risky scenarios	142
5.3.1	A punctual incident	142
5.3.2	Seasonal events	143
5.3.3	An emerging risk	144
5.3.4	No chronological pattern	146
5.4	<i>timePlot</i> in use	147
5.5	<i>timePlot</i> in misuse	148
5.5.1	<i>timePlot</i> used as a full text search engine	149
5.5.2	Filtering similarity and modelling aspects of scenarios	150
5.6	Lessons learned	151
5.7	Chapter conclusion	152
6	Dimensions of similarity: from simple lexical overlap to interactive faceting and multilingual support	153
6.1	Chapter introduction	154
6.2	Filtering aspects of similarity based on metadata	155
6.2.1	Overlap between textual similarity and coded data	155
6.2.2	Smoothing the major facets	157
6.3	Computing interlingual similarity	159
6.3.1	Constructing the pivots	161
6.3.2	Indexing the documents	163
6.3.3	Evaluation in a multilingual context	163
6.3.4	Discussion	164
6.4	Topic Modelling applied to the ASRS database	165
6.4.1	Topic Modelling in a nutshell	165
6.4.2	Interpreting the topics	167
6.4.3	Discussion	169
6.4.4	Application to similarity	170
6.4.5	Application to uncoded collections	170
6.5	Active learning for interactive model construction	170
6.5.1	An interactive approach to signal detection	171
6.5.2	Active learning algorithm	172
6.5.3	Simulation and results	174
6.5.4	Using <i>Dimensions</i> to produce KPIs: a use case	176
6.5.5	Possible application to similarity	176
6.6	Lessons learned and future work	177
	Conclusion	181
	Bibliography	191

List of Figures

1.1	The process of system failure	28
1.2	ICAO's Annex 13 official definitions of reportable events.	30
1.3	The failure type pyramid	31
1.4	Stages of incident reporting	33
1.5	Hierarchical model of risk management	35
2.1	Types of actors in civil aviation	44
2.2	Table of contents of NTSB/AAR-14/01	48
2.3	Excerpts from the "Factual Information - History of Flight" (§1.1, p. 19) section of NTSB/AAR-14/01	49
2.4	Excerpts from the "Factual Information - Personnel Information" (§1.5, p. 33) section of NTSB/AAR-14/01	50
2.5	Excerpts from the "Analysis - Flight Crew Performance" (§2.5, p. 86) section of NTSB/AAR-14/01	50
2.6	Description of the "low airspeed alerting system" system (§2.7, p. 104) of NTSB/AAR-14/01	51
2.7	Probable Cause (§3.2, p. 147) of NTSB/AAR-14/01	51
2.8	Some of the recommendations (§4, p. 148) of NTSB/AAR-14/01	52
2.9	Narratives form CADORS accident briefs (nu 2015P0532)	55
2.10	FAA AIDS Report nu 20140213000969I	56
2.11	ASRS monthly report intake	57
2.12	Synopsis and narratives of ASRS ACN1002555	58
2.13	Narrative of ASRS ASN45677 using the old writing style	59
2.14	Examples of reports from a SMS system	61
2.15	Report from ASCEND data feed	62
2.16	Report from The Aviation Herald	63
2.17	Report from ASN	64
2.18	Size comparison of reports from various sources in number of words	66
2.19	Metadata in ASN	72
2.20	Occurrence information in CADORS	73
2.21	Metadata of ASRS ASN1002555	74
2.22	The bow-tie accident model	76

2.23	ECCAIRS GUI	77
2.24	Sequence of events in ECCAIRS for the Concorde crash	79
2.25	Probable causes section of BEA report on the Concorde crash	79
2.26	ASRS search GUI	86
2.27	Fatal air accidents in France since 1987	87
2.28	General aviation accidents in France in 2013	88
3.1	The IR process	101
3.2	AH report 43028227	104
3.3	AH report 42d059bc	104
3.4	AH report 4317a108	105
3.5	AH report 42cbc93c	105
3.6	AH report 439534d5	106
4.1	Variants of MTOW	122
4.2	A processing chain	123
4.3	Synopsis of ASRS ASN796443	124
4.4	Synopses of ASRS ASN796443, ASN356951 and ASN241893	128
5.1	<i>timePlot</i> GUI: source report selection <i>via</i> search	139
5.2	<i>timePlot</i> GUI: source report selection <i>via</i> direct input	140
5.3	<i>timePlot</i> GUI: interactive scatter-plot	141
5.4	<i>timePlot</i> GUI: pop-up dialog of a similar report	142
5.5	<i>timePlot</i> GUI: Example of seasonality - bird strikes	144
5.6	<i>timePlot</i> GUI: Example of emerging risk - laser pointers	145
5.7	<i>timePlot</i> GUI: Example of a query with no pattern	147
6.1	Cross-lingual ESA	161

List of Tables

2.1	External typology of occurrence report corpora	65
2.2	Internal typology of occurrence report corpora	69
2.3	ADREP Occurrence categories	81
2.4	Typology of taxonomies	83
3.1	Examples of ADREP occurrence categories	116
3.2	Detailed scores per occurrence category	117
4.1	Features extracted from ASRS ASN796443	125
4.2	Features extracted from ASRS ASN796443	129
6.1	Highly associated terms for each class	157
6.2	Filtered and unfiltered mean overlap between textual similarity and coded data	158
6.3	An incident report from the CADORS database	164
6.4	Mate retrieval results	164
6.5	The 5 first topics extracted from the ASRS corpus	167
6.6	Results for <i>bird-strike</i> (DGAC corpus)	175
6.7	Results for <i>confusion</i> (ASRS corpus)	175

List of Acronyms

ADREP	Accident/Incident Data Reporting
AIDS	Accident and Incident Data System
ASRS	Aviation Safety Reporting System
ATC	Air Traffic Control
BEA	Bureau for Safety Investigations and Analysis for Civil Aviation (Bureau d'Enquêtes et d'Analyses pour la Sécurité de l'Aviation Civile)
CADORS	Civil Aviation Daily Occurrence Reporting System
CIFRE	Industrial Agreements for Training Through Research (Conventions Industrielles de Formation par la REcherche)
CL-ESA	Cross Lingual Explicit Semantic Analysis
DGAC	Directorate General for Civil Aviation (Direction Générale de l'Aviation Civile)
EASA	European Aviation Safety Agency
ECCAIRS	European Co-ordination Centre for Accident and Incident Reporting Systems
ESA	Explicit Semantic Analysis
ETL	Extract Transform Load
FAA	Federal Aviation Administration
GUI	Graphical User Interface
ICAO	International Civil Aviation Organisation
IR	Information Retrieval
KPI	Key Performance Indicator

NASA National Aeronautics and Space Administration

NLP Natural Language Processing

NTSB National Transportation Safety Board

PMI Pointwise Mutual Information

PPMI Positive Pointwise Mutual Information

R&D Research and Developpement

TSB Transportation Safety Board of Canada

Introduction

The first accident report

On one cold evening a very long time ago, shortly after our ancient ancestors had discovered the virtues of fire, in an attempt to quickly heat-up his dwelling one of them overstocked the fireplace with some particularly dry wood and struck a flint. At first all went well. The interior of the hut got cozy and warm. But the flames kept getting larger and larger. All of a sudden an ember sprung out and landed on the straw-lined floor of the hut. Another one landed on the stockpile of firewood close by. Both caught fire. Something was wrong. After an unsuccessful attempt to put the fire out, the man fled in panic and helplessly watched his home burn to the ground. The tribe gathered and, with the charred remains of the hut still looming in the background, the man started recalling his story to his puzzled audience. The man did his best to describe the events. The fire, the stockpile of firewood, the horror he just went through. They looked up to the full moon, it was the day the spirits of the dead came out. Something must have upset them. Then another member of the tribe recalled that, just the other day, something not much unlike this had happened to him. An ember had landed on his own stockpile of wood but he had managed to take it off just in time. A young boy recalled that earlier that day he had stumbled upon a pale green lizard and had kicked it into the river. The elders of the tribe united in a nearby hut and pondered on the situation for some time, trying to make sense of it all. Finally they came up with a solution. In order to appease the spirits, the tribe shall gather some food and, on the next full moon throw it in the fire. Pale green lizards were declared sacred and are to never be bothered again. Just in case the former didn't work, wood shall be stockpiled as far as possible from the huts' fireplaces.

That day the first accident caused by man’s poor understanding and improper use of complex technology had occurred. The first accident report had been submitted. The first post-accident investigation had been conducted and the first set of safety-related regulations had been issued.

Today millions of fires burn around the globe, breathing life in the immense apparatus that keeps our society on its feet. Complex techno-social systems such as energy extraction and production, transportation, healthcare, manufacturing and the military often involve thousands of individuals working with complex machinery, channelling vast amounts of energy. We take it for granted that these systems almost never fail. We entrust our lives on to while expecting them to forever innovate and outperform. Yet, safety is not a natural byproduct of the industrial process. In order to achieve and maintain acceptably low levels of failure, modern systems rely on a framework of regulatory processes that guide day to day work practices and decision making. And, as much as energy is the lifeblood of any industry, information is the vital fluid of its immune system.

Knowledge about past events is one of the primary components of safety management. One simply wants to know as much as possible about what has already happened in order to anticipate what might happen and prevent it.

From all the high-risk industries, civil aviation is the one that takes information about past events most seriously. When accidents occur, an inquiry is made to analyse its causes and provide recommendations to prevent the same accident from occurring in the future. Programs aimed at collecting data about incidents are in place from the 70’s. Efforts at standardising information about past events have produced complex accident models, taxonomies and dedicated software for safety experts working with such data. Information is shared and disseminated, making it relatively easy to obtain.

Of this information, most still circulates in the form of texts. From simple narratives jotted down by a stressed-out pilot on an i-pad to accident reports compiled by a committee after a year long investigation, incident and accidents reports consist mostly of natural language accounts. Due to the current modes of collection, these parts are practically unusable once stored in a database, yet safety experts are unanimous that, despite the normalisation and coding efforts, the texts contain valuable information, which when accessed helps them gain insights on the current state of the system.

Given the ubiquity of natural language in incident and accident reports, NLP¹ methods and techniques provide potential solutions to a variety of issues. Some, such as providing basic full-text search capabilities are easy to implement, but in order to apply them correctly, one needs to take into account not only the specific needs of the industry but also the specific characteristics of the textual material, such as its particular writing styles and

¹Natural Language Processing

the heavy use of domain specific vocabulary. For such solutions to be useful, one also needs to take into account the redundant information and overlap between taxonomies and natural language descriptions. Likewise, given the widespread use of taxonomies, text categorisation, a well proven technology, is applicable to incident and accident data and has the power to reduce the need for manual coding, while increasing the coverage of industry-standard metadata throughout a given collection. This requires however a thorough understanding of the specificities of these nomenclatures in order to correctly define the classification task.

In this thesis we take a computational linguist's perspective and look at the subject, using the textual data as a starting point. In such, our objective is twofold:

- We first explore the information involved in managing safety in complex techno-social systems, how the information is produced, conveyed, stored, aggregated, transformed, used, misused and sometimes lost as it flows through the regulatory framework of civil aviation. In doing so, we redefine the place occupied within this flow by the basic building block of information - written text.
- Having identified the needs of safety experts and explored the informational landscape in which they engage in their activities, we show how NLP can contribute to improving the tools used by them when working with incident and accident reports stored in electronic format and incorporate language processing in tools specifically tailored to the industry's requirements.

We show that text can be viewed not only as the vehicle of information between humans, but also as a resource that, when properly tapped and exploited has the potential to improve the overall quality of communication of safety-related information within a given system. And we show that by considering the specificities of the data and the sector, one both improves the quality of NLP applications designed to operate within the specific domain, better chooses specific NLP methods and technologies and better adapts them to the precise needs expressed by the community.

Background, context, evolution and achievements of this thesis

This thesis was launched in September 2010 in the form of a private-public research partnership between *CFH - Safety Data*², a small enterprise based in Toulouse and *CLLE-ERSS*³ research institute, part of *University of Toulouse - Jean Jaurès* and the *CNRS*, with backing from the French state through a CIFRE⁴ program.

CFH - Safety Data at the time were (and still are) working with a number of actors from civil aviation, both public entities such as the French state regulator (DGAC⁵), the authority responsible for carrying out safety investigations (BEA⁶), the European Aviation Safety Agency (EASA⁷) as well as private aircraft manufacturers and service providers.

They were developing a text-based document classification solution for incident and accident reports (Hermann et al., 2008) and were seeking to expand their R&D⁸ effort to more than just classification of accident reports. This project was thus launched in partnership with the NLP group of *CLLE-ERSS*, with an initial focus on prediction and identification of human-factors related issues in free text with the objective to integrate such functionalities in *CFH - Safety Data*'s existing commercial solutions destined at safety professionals.

At that time, with the usual enthusiasm associated with the beginning of a thesis, we focused the initial research project around the idea of “weak signal detection” and our goal was to propose methods for identifying new and unseen risky scenarios in incident and accident report narratives. We started looking at methods for detecting outliers and statistical anomalies. As some of these methods are based on distance (or similarity), we started playing with document-document similarity and very early on (winter 2010) we proposed a basic application for identifying similarities among incident and accident reports. In order to present these reports, rather than showing a list of documents to the user, the application would make use of an interactive visualisation technique, combining chronological distribution and textual similarity. When we presented the prototype to the clients of *CFH - Safety Data* data they immediately found it very pertinent to their everyday needs. The prototype became the *timePlot* system, which we present in Chapter 5.

²<http://www.safety-data-analysis.com/>

³<http://w3.erss.univ-tlse2.fr/>

⁴Industrial Agreements for Training Through Research (Conventions Industrielles de Formation par la REcherche)

⁵Directorate General for Civil Aviation (Direction Générale de l'Aviation Civile)

⁶Bureau for Safety Investigations and Analysis for Civil Aviation (Bureau d'Enquêtes et d'Analyses pour la Sécurité de l'Aviation Civile)

⁷European Aviation Safety Agency

⁸Research and Développement

In all honesty the commercial success of the system astounded us as we quickly discovered that the domain of aviation safety was particularly ill-equipped for dealing with large amounts of textual information. While everyone was accustomed to using powerful search engines in their everyday activities, the applications used by safety experts still relied on decades-old boolean retrieval methods. Partly driven by the commercial success of the *timePlot* system (and the need to develop and maintain it), and partly by our discovery, that there was a clear need for tools allowing easy access to the textual information, we shifted our focus on exploring the needs of *CFH - Safety Data*'s clients in their daily uses of incident and accident reports.

We found out that access to information in databases of incident and accident reports relies primarily on hand coded metadata attributes and that due to a number of factors, these are not always reliable. Sometimes they aren't available at all. This creates a particularly frustrating situation, where information about life threatening issues hides inaccessible in masses of unstructured data, while it continues to pile up by the thousands of reports due to a monumental effort, made in order to collect information about potentially life threatening issues.

So, as computational linguists, we compared large amounts of incident-data from a variety of sources, both public and private and drew a panorama of the current collection, exchange, storage and analysis paradigms. We identified a significant gap between the intentions of the systems' designers and the reality of the data, particularly when it accumulates over time. Even more, when considering data exchange between institutions, a current topic of interest, format incompatibility hinders free information flows within the system. Considering the current top-down approach of mapping data to static pre-established taxonomies, we propose a robust bottom-up method that, in our opinion has the potential to complement or (in time) even replace certain aspects of the established data codification strategies.

The *timePlot* system proved to us that we were on "the right track" from a user point of view and prompted us to further refine the needs, this time based on observations of the actual use of the system and on conversations and interviews with its users. This lead *CFH - Safety Data* to invest into a production grade system destined specifically at incident and accident reports.

At the same time we continued to explore the notion of textual similarity and played with some of the "hot" current technologies, such as *Topic Modelling* and explored the redundancies between taxonomies and natural language narratives. Having proven the usefulness of textual similarity in the context of data driven risk management, we explored its different facets and complexities, questioning for example the unidimensional character of the (similarity) scores compared to the inherent multidimensionality of the data. We also faced more practical issues, such as multilingual data sets and propose robust methods for language independent similarity modelling.

Also, working with the feedback from the document-document similarity based methods we identified the limitations of the proposed approach, such as the need to provide a single document as a point of entry. Building on the experience we propose a method and process, allowing the user to directly model and project a given aspect (or dimension) of interest on the data, using iterative machine-learning for control and validation of the results.

Document outline

This document is organised as follows:

Chapter 1 introduces the basics of accident prevention. We first discuss the events that need to be studied in order to prevent accidents. Next, we take a look at safety as a multidisciplinary and broad problem, ranging from airplanes to politicians and how information about incidents plays a key role in managing it. Finally, through an example, we discuss how this information is used.

Chapter 2 presents incident and accident data and how it circulates through the regulatory framework of civil aviation. We will start by applying a risk management model to the sector and list the different entities involved in the safety process. Next, we will see a representative cross-section of the types of occurrence data, how it is produced and what information it vehicles. Next, we will explain how this data is stored and organised using *taxonomies*, before showing examples of how this data is used in order to improve the safety of civil aviation and discussing what the main problems that arise when manipulating occurrences on a large scale are. Finally we draw up a list of needs can be addressed by NLP applications.

Chapter 3 presents the domains of Information Retrieval and Text Categorisation and how they answer the needs expressed by the aviation safety community. Each section is organised by first presenting the domain and the key concepts, before discussing the specific implication of their application to occurrence data.

Chapter 4 is divided in two parts. First, we present our solution to the problem of normalising the textual material we encounter in incident and accident reports in order to transform it to formats suitable for vector space modelling. Next, we discuss the vector space modelling framework, central to many current NLP methods.

Chapter 5 presents the *timePlot* system for detecting similar occurrence reports. We present the tool's graphical interface and show examples of the results it presents to the users. We then discuss how the tool was really used and how, by observing the actual use of such a tool, we came to gain further insight into the needs of the users.

Chapter 6 explores the notion of *similarity* from several different angles, each addressing a different aspect of the complex notion. We first present a method that learns from documents and their associated metadata attributes and allows to filter out one or another *facet* of similarity. Next, we address the question of multilingual databases and explore the potential of second-order similarity methods to model collections of documents written in different languages. Next, we compare the results of *Topic Modelling* to the information in ASRS's metadata and study their overlap. Finally, we present an approach based on *active learning*, allowing a user to model a certain aspect of an accidental scenario by providing the system with a few initial examples.

Chapter One

Basics of accident modelling and risk management

“Pay attention, that’s all,” Eliza said. “Notice things. Connect what you’ve noticed. Connect it into a picture. Think of how the picture might be changed; and act to change it.”

— Neal Stephenson, *The Confusion*

In this chapter we introduce the basics of accident prevention. In Section 1.1 we define the events that need to be studied in order to prevent accidents. Next, in Section 1.2 we take a look at safety as a multidisciplinary and broad problem ranging from airplanes to politicians and how information about incidents plays a key role in managing it. Finally in Section 1.3 we briefly discuss how this information is used by safety experts.

1.1 What is an accident?

Today we all seem to think we know what an industrial accident is. Industrial accidents are those events when we lose our mastery over technology we built ourselves, when things get out of control and cause damage, injuries or death. We are all familiar with the scenes of utter devastation immediately following a plane crash or a factory explosion as they are favourite prime-time material for the media. Burning wreckage and grieving victims' relatives fill the TV screens and grip the public's mind. The media debate in the immediate aftermath is quick to recall other similar¹ accidents from the past, instituting a (fortunately temporary) sense of helplessness in the face of the rapidly advancing technology and inevitably leading to a debate about the (un)safeness of whatever system happened to fail.

In a way we are right. Spectacular crashes involving loss of life are extremely good examples of accidents. Fortunately they are also extremely rare and isolated events. In 2012 the probability² of dying on a single flight on one of the top 39 airlines was one in twenty million, roughly that of winning the French national lottery with a single ticket. During the last decade³ nearly a third less lives were lost in aviation accidents worldwide than in traffic accidents in Bulgaria, a country home to about 0.1% of the world population. Also, safety in air travel is constantly improving. ICAO⁴ reports 2013 as the year having the lowest accident rate (2.8 accidents per million departures) since they started keeping the record (ICAO, 2014). The system is becoming safer. In the vast majority of cases, even when something serious, such as an in-flight engine malfunction happens, the accident is avoided and the aircraft lands safely. Even more often something could have happened but was avoided in time. Today when two airplanes are on a collision course, on board automated systems detect the danger well in advance and advise pilots on the appropriate evasive maneuver. Finally, even after an accident occurs, one's chances of survival are much greater today than several decades ago, due to ever improving cabin design, passenger evacuation and ground emergency procedures.

1.1.1 From normality to disaster

So what is an accident? According to Hollnagel (2004):

¹By similarity, the media seem to think that the outcome in terms of death tolls is more important than the comparability of the events themselves.

²Source: OAG Aviation & PlaneCrashInfo.com accident database, 20 years of data (1993 - 2012)

³From 2003 to 2013, 5085 perished in Aircraft accidents, while over 8400 deaths are recorded in the official Bulgarian statistic for traffic accidents.

⁴International Civil Aviation Organisation

[A]n accident can be defined as a short, sudden and unexpected event or occurrence that results in an unwanted and undesirable outcome. The short, sudden, and unexpected event must directly or indirectly be the result of human activity rather than e.g., a natural event such as an earthquake. It must be short rather than slowly developing. The loss of revenue due to an incorrect business decision can therefore not be called an accident, regardless of how unwanted it is. It must be sudden in the sense that it happens without warning. The slow accumulation of toxic waste in the environment is not considered an accident since in this case the conditions leading to the final unwanted outcome - the disruption of the ecology - were noticeable all along.

Accidents leading to loss of life, injury or property damage are however just the most extreme manifestation of a whole spectre of events that we can order from most severe to least severe: events where something went wrong, events where something could have gone wrong or simply events where in one way or another things didn't work out as we expected them to.

Considering the above definition as a basis for prevention, Hollnagel also points out that an accident is the coupling of an event and an outcome and that the end goal of accident prevention is ensuring that "recipient comes to no harm", in other words avoiding the outcome, even if the event itself is unavoidable.

The high level process-oriented view of a developing accident, provided by C. W Johnson (Johnson, 2003) using Turner's model of system failure (Turner, 1992), allows us to better understand how a catastrophe develops.

Initially the system is in a *normal state*. During an *incubation period* conditions, such as undermaintained components or inappropriate work practices gradually build up, rendering the system accident-prone. (Flammable material is stored next to a potential ignition source). The accident is waiting to happen. A *triggering event* (an ember flies out of the fireplace) causes failure and triggers a sudden and unexpected event (The straw on the floor ignites). As the event unfolds, it may trigger a set of other sudden unexpected events and the situation may escalate rapidly (The small fire grows large, the furniture starts burning, the hut collapses.) A chain reaction leads to disaster. The loop on the left of Figure 1.1 illustrates this chain reaction. Immediately after the event, efforts are made to *mitigate* the failure, (The small fire is rapidly put out) returning the system to a normal state. This is illustrated by the loop on the right of Figure 1.1. Actions are taken to return the system to a normal state. During *salvage and rescue*⁵ the system physically recovers from

⁵Including *rescue* in this stage creates, in our opinion, some overlap with the mitigation phase as the rescue effort is part of damage control procedures and thus limits the severity of the outcome.

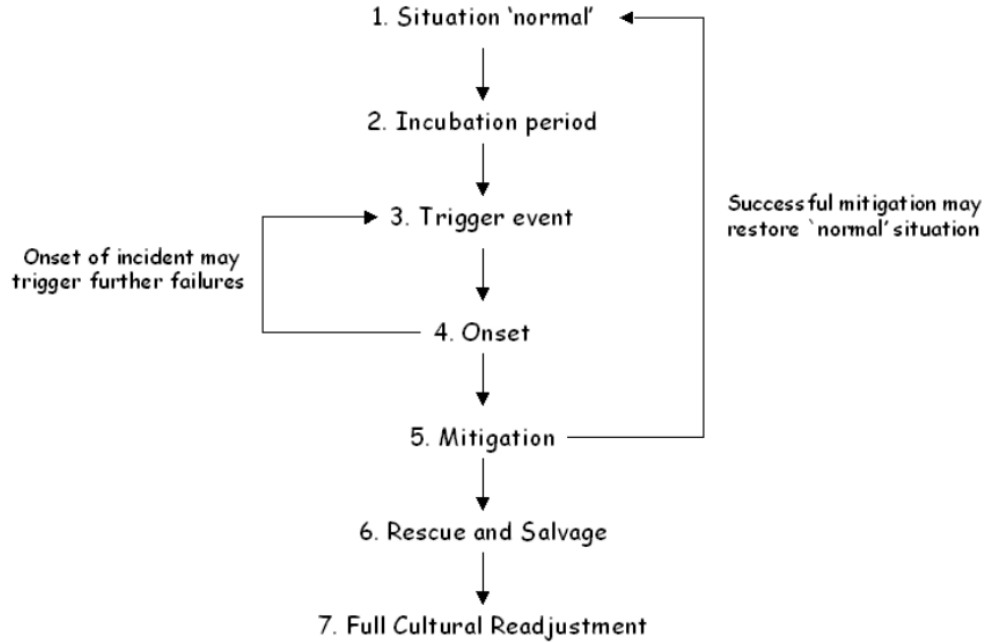


Figure 1.1: The process of system failure

consequences of the event and at the final stage, *cultural readjustment*, lessons are (hopefully) learnt from what just happened.

An accident can therefore be described as a complete instantiation of the process from stage 1 through 7. The magnitude of the accident depends on the mitigation measures taken in order to stop a minor event from escalating. Stopping before the situation gets out of control signifies breaking the chain reaction between the onset (stage 4) and a new triggering event (stage 3).

The (initial) triggering events, however, can be so insignificant that immediate mitigation measures are so effective that the accident is “stopped in its tracks” and phases 6 and 7 do not occur. Furthermore, as we will see, a properly functioning system by definition does not allow development further than phase 2. Gradual build-up of the very conditions potentially leading to an accident is not allowed.

In any case accident prevention is largely about monitoring and understanding the system. When an accident occurs, crucial lessons are learned. Those lessons are applied to the system so the process of failure is interrupted at the earliest possible stage. In civil aviation, official accident investigations (§2.1.2) produce this kind of feedback. However, as the system gets safer, we (fortunately) have less and less concrete examples of accidents to work with.

In order to improve safety, we therefore have to work with cases, where the failure process was initiated but was stopped before becoming a full-blown catastrophe: incidents and abnormal situations. Monitoring such events becomes the focal point of the ongoing effort of improving the safety of an *almost* perfect system.

1.1.2 A complicated definition

We saw in the previous section how an accident can be defined, but we also saw that there are events that do not fit into this definition. Defining them is no easy task. Seeking to discretise the spectre from the edge of normality to total devastation has led to a plethora of definitions. Hollnagel (2004), for example proposes four categories of events:

- **Accidents:** events resulting in death, injury and/or serious property damage
- **Incidents:** events having an unwanted outcome, that had the potential to progress to accidents.
- **Near miss events:** Events without an unwanted outcome, that had nonetheless the potential to become incidents or accidents
- **Unsafe acts:** Events that almost reached the threshold of near miss events

In the same spirit, in civil aviation, events which should be reported are split in three categories. Annex 13 to the Convention on International Civil Aviation (ICAO, 2001) gives the definitions shown in Figure 1.2.

In ICAO's definition a continuum is clearly present. An accident is defined as a function of the severity of the occurrence. An *incident* is defined as opposed to an *accident* and a *serious incident*, defined as an incident that was *almost* an accident, manifesting the overlap between the two concepts.

Johnson (2003, pp. 17-18) perfectly illustrates the difficulty of defining events ranging from, say, the discovery of an apple on the floor of a jetliner's cockpit⁶ to the meltdown of Chernobyl's reactor core. He gives a meta-definition, listing seven different *strategies* at attempting to define these events.

It is not in the scope of this thesis to discuss the conflicting definitions of what an accident or incident is, nor to provide yet another one. So, in order to skirt the accident/incident dichotomy and the need to discretise what is clearly a continuum, we will employ the term **occurrence** and define it in a

⁶This particular event comes from the internal incident reporting program of an airline. The danger is that the fruit may block the rudder pedals at an inappropriate moment during the flight.

- **Accident:** An occurrence associated with the operation of an aircraft which takes place between the time any person boards the aircraft with the intention of flight until such time as all such persons have disembarked, in which:
 - a) a person is fatally or seriously injured as a result of:
 - * being in the aircraft, or
 - * direct contact with any part of the aircraft, including parts which have become detached from the aircraft, or
 - * direct exposure to jet blast,

except when the injuries are from natural causes, self-inflicted or inflicted by other persons, or when the injuries are to stowaways hiding outside the areas normally available to the passengers and crew; or
 - b) the aircraft sustains damage or structural failure which:
 - * adversely affects the structural strength, performance or flight characteristics of the aircraft, and
 - * would normally require major repair or replacement of the affected component

except for engine failure or damage, when the damage is limited to the engine, its cowlings or accessories; or for damage limited to propellers, wing tips, antennas, tires, brakes, fairings, small dents or puncture holes in the aircraft skin.

or
 - the aircraft is missing or is completely inaccessible.
- **Incident:** An occurrence, other than an accident, associated with the operation of an aircraft which affects or could affect the safety of operation.
- **Serous incident:** Serious. An incident involving circumstances indicating that an accident nearly occurred.

Figure 1.2: ICAO's Annex 13 official definitions of reportable events.

very open manner.

An occurrence is an observable manifestation of a deviation from normality within a given socio-technical system.

At this stage we choose to qualify the deviation as “observable”, rather than observed, because the very notion of the observer may vary as will be seen in the following sections. An occurrence may, for example, be observed by someone but, as is often the case, the information may not be available to whoever is in a position to correct it.

The notion of normality is to be taken here at face value. A system functions normally when it does everything as it is intended to. The frontier between normal and abnormal is fuzzy at best, but there is still much to be done in areas that are clearly on the abnormal side and away from the fuzziness.

1.1.3 Severity, frequency and visibility

Now that we saw the types of events that interest us, let us look at how they are distributed. Working on industrial accident prevention in the 1930s, safety science pioneer William Heinrich compared thousands of accident reports and came up with a ratio, stating that for every accident causing major injury, there are 29 accidents causing minor injuries and 300 accidents causing no injuries (Heinrich et al., 1980).

A similar study done by Bird (1984) came up with a 1:10:30:600 ratio (commonly illustrated as seen in figure 1.3) studying close to two million industrial accidents.

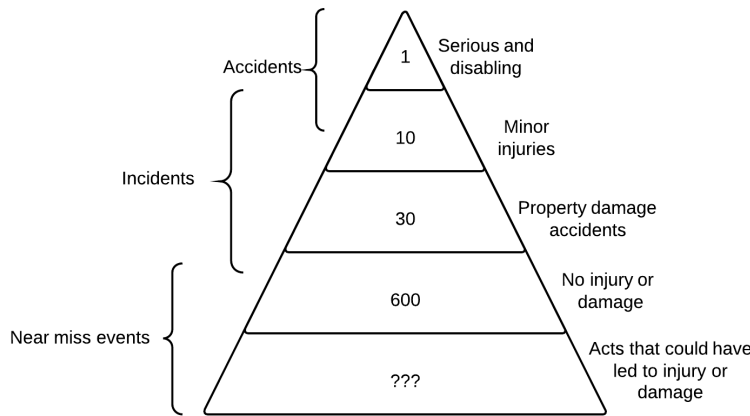


Figure 1.3: The failure type pyramid

Known as the Heinrich Pyramid, these ratios illustrate a relationship of inverse proportion between the severity of events and their frequency. The exact ratio varies between industries, but the relationship always stands. For every death, there are many more injuries. For every injury there are many more incidents resulting in no injury and for every such incident there are many more near miss events, where the incident was completely avoided.

We will not discuss the (mis)uses of this model for statistical prediction of injury based on reported occurrences, nor the actual ratios reported by Heinrich or Bird (Manuele, 2013). For our purposes, the pyramid is useful as a purely theoretical construct. Every occurrence can be placed somewhere on the pyramid. With the fuzzy frontier of normality at its very base it follows that, as a whole the pyramid contains all the information about every

deviation from normality in a given system.

What is important is that in reality there is also an inverse relationship between severity and visibility. We can be certain about the number of accidents. Failures with lesser consequences are reported with less consistency and reliability. Near-miss events are only rarely reported (Hollnagel, 2004).

The iceberg metaphor is more than tempting. In a natural state, without any effort, Heinrich's pyramid could be thought of as an iceberg. At the very top, visible from afar, are the spectacular disasters we all know. The information about them is accessible and available. However the main body of the information about abnormalities in the system lies beneath the water-line and is not accessible without taking concrete measures to make it visible. And the further we dive, the darker it gets.

Gaining access to this information boils down to providing a feedback channel to handle the data about minor occurrences and creating the social environment and favourable conditions for the production and use of the data. The practice of organised collection of data about abnormal events is known as incident and accident reporting.

1.1.4 The basics of incident reporting

Gathering data on adverse occurrences is done through *incident reporting* programs. Individuals in the workforce are incited to share information through an established feedback channel. Throughout industries, such programs are becoming more and more common. In civil aviation reporting is mandatory (ICAO, 2001).

Programs vary in size scale and complexity, from highly specific local endeavours to national programs involving multiple actors across institutions. Their ultimate aim however remains the same: to identify the causes of previous failures and to use this understanding to avoid or reduce future problems. (Johnson, 2003).

Figure 1.4 shows the generic process of gathering information about adverse events (Johnson, 2003).

When an event occurs, it must first be *detected* and an initial notification produced. Follows a *data gathering* stage during which relevant information (facts) about the occurrence are collected. Follow *reconstruction* and *analysis* phases during which the incident scenario is developed and the causes of the failure identified. Building on this information, *recommendations* about how to act and change the system are produced. Finally this information is shared with other interested parties.

Information is produced and transformed at all stages of the process by different actors. How exactly depends on the specific implementation of the incident reporting program. As they vary in scale they also vary in complexity. Johnson discusses in detail different implementations of such programs (Johnson, 2003, ch. 4) and we will see examples in the next chapter.

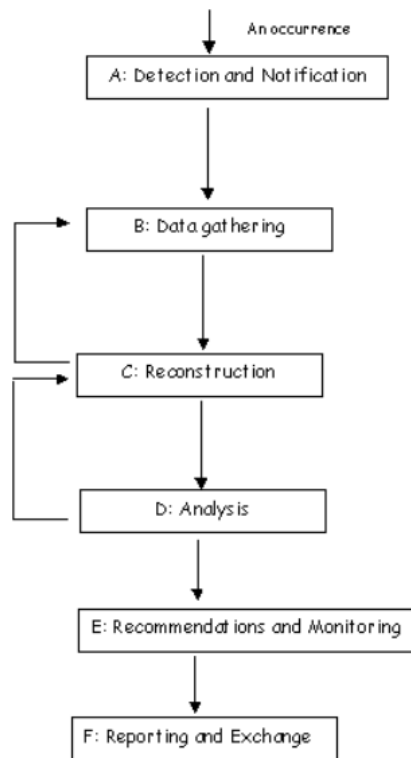


Figure 1.4: Stages of incident reporting

Whatever the specific implementations, all such programs have in common that they are in effect active channels of communication. Information about occurrences circulates in them mostly in the form of electronic documents⁷. While the occasional phone call and face to face interaction are integral part of incident reporting, the information that they generate is used either to create a new document or to complement an existing record.

At the beginning of the incident reporting process a detected occurrence is reported in most cases by the person who experienced it in the form of a short narrative combined with summary factual information. The report is often filed using standardised forms in paper or electronic format.

When the initial report is received, it is processed and, depending on the specifics of the implemented architecture, more or less normalised. It is also at this stage that, based on how safety-critical the occurrence is, the report may be routed for immediate attention and further processing or just stored.

⁷By documents here we refer to any piece of electronic matter that contains information or evidence.

The bulk of information is generated during the reconstruction and analysis phase. At this phase software solutions such as ECCAIRS⁸ (§ 2.2.3.5) assist the analysts by providing a description framework and a work flow to assist the process. The textual narratives are complemented with predefined attributes containing the required factual data (§ 2.2.2). The occurrence is *coded*. However, in some cases this stage is skipped altogether for non-critical occurrences and they are just stored in the collection.

The last phase of the reporting process involves sharing the information. In the next section we will see how this information fuels the risk-management process. Information shared at this stage also constructs the body of available knowledge and data to which practitioners are free to turn and look whenever safety-critical decisions require evidence to back them up. It is also at this stage that information gets fragmented and tools such as aggregated databases with powerful search and coherent classifications have the potential to assist experts working on this body of information.

1.2 Risk management in a complex systems

In order to better understand how such information fuels the risk management process as a whole, we can turn to the model proposed by Rasmussen et al. (2000). In his view accidents are caused by dysfunctions at every level of a complex and far-reaching interconnected system. In other words, decisions taken by pilots as well as politicians impact the overall safety of the system.

In this model, risk management is viewed as a problem of controlling work in order to avoid loss of control over physical processes. The model is a hierarchy, ranging from government, companies, management through to systems operators. Figure 1.5 (Rasmussen, 1997) provides an example, although the exact number of levels may vary across industries.

Starting from the very immediate proximity to the work practice, Rasmussen's "ladder" represents the system as a series of levels where each higher level has some form of control over the lower one.

The lowest level, L6 concerns the **technology** itself - how equipment is designed and how the operating procedures for the equipment are communicated and understood. Level L5 concerns the activities of the **individuals** that interact directly with technology. Level L4 concerns management - how staff is controlled and how work is organised. Level L3 concerns the activities within a given *company*, how regulation is understood and applied within its perimeter. Level L2 concerns the activities of various *regulators*, whose task is to implement the legislation in their respective sectors. Finally, level L1 concerns the activities of **government**, crafting legislation that controls the practices of safety in society.

⁸European Co-ordination Centre for Accident and Incident Reporting Systems

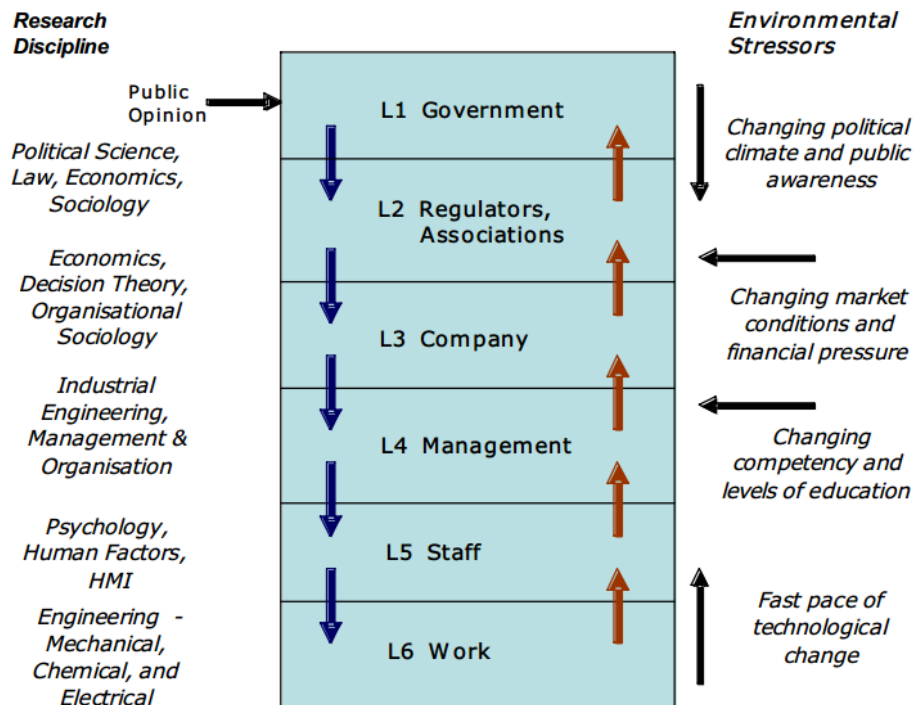


Figure 1.5: Hierarchical model of risk management

In a dynamic and constantly evolving world, the different levels are subject to various disruptive forces to which the system must adapt. L1 for example is influenced by shifts in public opinion and politicians seek to respond by changing legislation. On the level of individual companies, changes in the market such as competition or shortage of resources call for counteraction. On the level of staff and (L4, L5) management, phenomena such as “normalisation of deviance” (Vaughan, 1996) introduce a gradual and continuous shift towards riskier behaviours. Finally the ever more rapidly evolving technology causes constant changes to level L6.

Understanding each level involves different academic disciplines: political science (L1, L2), law (L1, L2), economics and sociology, (L1, L2, L3), organisational psychology and management theories (L3, L4), human-machine interaction and human factors (L5) and various engineering disciplines (L6). Change affects the system as a whole, but radically different frameworks are used to analyse and adapt to change at different levels of the hierarchy. This leads to misalignments that weaken the system and lead to catastrophes.

When changes are made at higher levels, they often disregard the implica-

tions that they have on the lower levels of the hierarchy. When changes occur on lower levels of the hierarchy, the higher ones must be adequately informed in order to adapt accordingly.

In order to ensure safe operations within the system, vertical alignment of the different levels must be maintained. This boils down to ensuring effective two-way information flow within the hierarchy.

1.2.1 The descending flow: controlling the processes

First and foremost, decisions taken on the higher levels must transmit adequately to the lower levels of the hierarchy. Political decisions taken by government (L1), must be translated to regulation (L2) and respect of regulation must be ensured through some form of control. The same goes to practices within a given company (L3). Safety-related decisions (such as new operating procedures) must be transmitted to management (L4) which must ensure that they are respected by operators (L5), whose actions on the controls must result in the expected behaviour of the equipment (L6). This descending flow of control forms the “backbone” of safety management.

Traditionally, the descending flow of control and regulation has been the subject of efforts to increase safety. In the last few decades, however the attention is shifting more and more towards the ascending one, that of process-levels information arriving and informing decision-making ones. The reasons for this shift of attention are multiple:

- Systems today operate on ever increasing scales. In a interconnected world it is not uncommon for a given industry to become continental or even global, as is the case with international rail systems or aviation. This brings the need for synchronisation of decisions on a much greater scale by introducing greater distance between operations and decision making, putting increasing demand on the existing feedback channels.
- Systems get more and more complex with time. With scale and advancing technology, the number of “moving parts” within a system, both in a strict sens and metaphorically speaking increases dramatically. There are millions of parts in a single airplane. In most cases they are produced by hundreds of subcontractors from all over the world. For a single flight to be completed thousands upon thousands of interactions need to be performed ranging from the pilot acting upon the throttles, through different air-traffic controllers ensuring a free corridor up to the airline personnel calculating fuel needs and even booking the hotel for the pilots. Each one of these interactions has the potential to impact safety and all need to be considered. With complexity the need for empirical data for decision making is increasing.

- Technology advances at an ever more rapid pace. The development cycles of products are becoming shorter and shorter. Innovations are becoming operational faster than before. As every change in the system has the potential to affect safety, the demand for adequate monitoring of the effects of these changes increases.
- Systems are becoming safer. With time and well functioning risk management, common and “simple” sources of failure are eliminated. In consequence, today’s accidents are of a far more complex and uncommon nature than those of yesteryear. Understanding and preventing them thus requires a far more detailed knowledge of the underlying processes and of the system as a whole (Amalberti, 2001).

1.2.2 The ascending flow: information driven decision making

In order to make adequate decisions at the higher levels, information about processes at the lower ones needs to propagate freely up the hierarchy. Decision makers can not operate “in the dark”, without knowledge of the system they are controlling. This goes for all levels of the hierarchy.

At the lowermost levels of the hierarchy, the immediate state of the system (L6) must be understood by the operators (L5) through an adequately designed human-machine interface. As a car’s dashboard must coherently present information such as speed and remaining fuel, an aircraft’s cockpit instruments or a nuclear power plant’s control-room must present all the relevant information to the operators.

In the same manner management (L3) and company (L4) must be kept informed by the operators (L2) on the current state of operations. Often work practices diverge significantly from official procedures. It is thus impossible for management to adequately control work without up-to-date knowledge about the reality of operations and feedback channels (internal incident reporting) inform management when the processes start to drift towards unacceptable levels of safety. At this level however information starts being produced by humans, rather than machines and its adequate consumption on the higher level is highly dependant on both the capacity of the channel and the interpretation techniques used at its reception to handle the inherent variability and instability of human communication. **Natural language, with all its imperfections becomes the only available interface.**

At the level of regulators (L2), changes on the lower levels must be thoroughly understood and monitored. At this level (and to some extent at the previous level in large companies), information overload starts being an issue. Feedback channels must not only be in place, but adequate aggregation, synthesis and signal analysis methods are needed to efficiently filter relevant from irrelevant information and cope with the increase of scale. *Data over-*

abundance becomes a problem.

Placed within Rasmussen's model, the bulk of the work in this thesis addresses issues situated on the ascending flow of information between the front line operators and the higher levels of the hierarchy. It is this information that safety experts need in order to gain insight on the overall state of operations.

1.3 Looking for patterns

When collected, information about occurrences is interpreted by safety experts looking for indications of a potentially risky situation.

Operations within any given system are of a cyclical nature and have a clearly defined perimeter. Air travel is ultimately about a whole lot of airplanes flying from point A to point B. Manufacturing is about transforming raw materials and making the final product over and over again. Healthcare is ultimately about transforming sick people into healthy people. In each case the process is repeating itself. This repetitiveness entails that similar failure processes are initiated again and again forming a pattern.

Occurrences therefore also are of an interrelated nature. When looking through incident reports, experts exploit their interconnected nature to gain insights and perceive risky scenarios. *Making patterns, drawing connections* and *perceiving novelty* are some tactics experts use when working with such data. Macrae (2007), observed safety experts at their work and gives the following two examples to illustrate these tactics:

Example 1: "An aircraft nearly used the full length of a runway to land.

The crew reported that on an approach in heavy rain they failed to override the automatic reverse thrust due to unrelated confusion over the apparent failure of a windscreen wiper. This event was immediately deemed [by the expert doing the analysis] "a bit of a QF1", referring to the flight code of another airline's aircraft that had overrun a runway and ended up in a field a few years previously. In that case, a water logged runway, poor crew communication and an inadequate braking technique were contributory factors. These factors were the basis of the connection drawn between that accident and this incident. This connection lead the investigator to suspect that a superficially inconsequential incident may point to an emerging and unrecognised problem in landing and approach discipline."

This account taken from Macrae (2007) shows how the nature of the expertise is manifested in the very creation of the relation. Drawing a connection between these events means first isolating the factors that were alike in both

incidents (runway overrun, water logged runway) factoring out elements such as the ineffective crew communication and filtering out factors such as the malfunctioning windscreen wiper, which was incidental in one of the events.

Example 2: “A series of events indicating improperly secured cargo

Over a couple months, investigators noticed several similar events involving pieces of cargo in the aircraft hold being found, on arrival, to be improperly fastened down, not fastened at all, or flight crew reporting hearing a ‘thump’ or ‘bump’ during flight. Unrestrained cargo can be a problem if it moves around and affects the trim and handling of the aircraft. Three events had been reported in the first month, and then it went up to about seven in the second month, and they had been seeing “bigger lumps of cargo” moving around into the bargain. Although the incidents themselves had little actual impact, investigators flagged this up as a “minor issue snowballing”. These events presented a clear pattern, suggesting that something was amiss in the loading of cargo. On closer examination, investigators found that all the events could be traced back to the same terminal, reinforcing and localising their suspicion of an underlying problem with work practices there.”

In this example, not knowing the existence of all of the cargo-related events would have obviously prevented the experts from making the connections. The opposite is also true. Having to keep track of hundreds or even thousands of parallel occurrences would exceed the capacity of any single human. Furthermore the final element needed to make the connection was extrinsic information - the identical localisation of all the occurrences. This example shows the importance of access to well organised and categorised databases.

In both examples interpreting the occurrences was based first and foremost on establishing a relation between them and categorising the nature of that relation. In both cases the reasoning was based on a thorough expertise of the domain and information about the occurrences. Investigators had knowledge about these events. In the first example it seems that the “QF1” occurrence was well known, and immediately came to mind. In the second example investigators probably became “alert” when several similar events were reported over a short period and were looking out for more of the same kind. Effectiveness of these tactics depends mostly on the expertise of investigators and providing them with just the right amount of information.

This type of reasoning pointed us to the basic need in the industry for facilitated access to information contained in incident and accident narratives and ultimately to the prototype presented in chapter 5 and the further research into the subject presented in chapter 6.

1.4 Chapter conclusion

We saw in this chapter how system safety depends on monitoring and analysis of incidents and accidents and that, in safer systems we do not have the “luxury” of learning from fully developed accidents and need to shift focus towards minor events. These events are however, by definition much more numerous and difficult to get at without effective feedback channels. We also saw that the issue of control of the risk-prone processes involves a complex of individuals and entities, ranging from pilots to politicians and how information ensures adequate decision-making at every level. It follows that, even when captured, due to the integral nature of safety prevention within a complex and interconnected system, such as aviation, effectively using this information becomes a problem of even grater scale. The availability of proper storage, processing, exchange and analysis mechanisms become a necessity.

At the end, however, most of the feedback data circulates in the form of uncontrolled natural language accounts of occurrences. As we will show, these accounts are today still treated as “dead weight” in the system and intended for human consumption only. It is our objective in this thesis to show that these texts can instead be considered as the raw input material to a series of automated processes that reinforce the aforementioned synthesis and analysis mechanisms already in place. For this, let’s now look at how safety is maintained within civil aviation and in particular at the data being produced, the solutions for its storage and the particular ways in which it is consumed.

Chapter Two

Safety information in civil aviation: actors, models and data

“I never saw a wreck and never have been wrecked, nor was I ever in any predicament that threatened to end in disaster. [...] I cannot imagine any condition which could cause a ship to founder. I cannot conceive of any vital disaster happening to this vessel. Modern shipbuilding has gone beyond that.”

— Cpt. Edward Smith (Captain of Titanic)¹

This chapter is about incident and accident data and how it circulates through the regulatory framework of civil aviation. We will start by applying a risk management model to civil aviation and listing the different entities involved in the safety process. In Section 2.1 we will see a representative cross section of the types of occurrence data and how it is produced. Next, in Section 2.2 we will explain how this data is stored and organised using *taxonomies*. We will introduce the concept of *meta data* and show examples of different solutions. Then, in Section 2.3 we will show how this data is used in order to improve the safety of civil aviation. Finally in Section 2.4 we will discuss what are the main problems that arise when manipulating occurrence on a large scale and how NLP solves some of them.

¹New York Times, April 16, 1912

A century of failures

As we saw in the previous chapter, when the system fails, light is shed on its inherent weaknesses and actions are subsequently taken in order to improve its robustness. Civil aviation is no exception, where significant accidents are also the most important catalysts for improving safety. Major changes are introduced to the system, in the wake of every plane crash, slowly shaping the manufacturing and regulatory landscape as we know it today. Before diving in the details of how data is produced, managed and used, let us take a look at three major early accidents that influenced aviation on the manufacturing and organisational levels.

On the manufacturing level, airplane design has been a continuous process of trial and error. Starting from the Comet Crashes in 1954 (RAE, 1954), meticulous investigation of accidents helped reveal the causes of countless technical failures and propose solutions and design improvements that are present in all of today's aircraft. The De Havilland Comet was the first commercial passenger jetliner and it was not before two of them exploded in mid air instantly killing all aboard, that they were declared unsafe to fly, due to a combination of poor design and manufacturing techniques.

Pressurisation-depressurisation cycles caused fatigue cracks to form at the corners of the airplanes' square windows. The cracks grew bigger and bigger until structural integrity was lost and the aircraft literally popped like a balloon. Today, due to the lessons learned from these accidents, mid-air explosive decompression due to metal fatigue is a thing of the past.

This particular series of accidents also has the merit to have founded the discipline of accident investigation itself. In the immediate aftermath, the United Kingdom saw its ambitions at becoming a global leader in commercial jet-powered aviation suddenly grind to a halt. Consequently, a considerable political will² was directed at finding the problem. Given that both aircraft had disintegrated at cruising altitude and over the Mediterranean sea, very little evidence to what went wrong was readily available. The investigators had to seek help from the Royal Navy in recovering the wreckage from the sea bed (a first) and then workout a theory to why the aircraft had exploded. The hypothesis gradually narrowed down to metal fatigue and in order to prove their theory, the investigators conducted a real-scale test by enclosing the same aircraft in a sealed water tank and subjecting it to endless pressurisation cycles until the fuselage lost structural integrity, thus proving the metal-fatigue theory. All aircraft with a pressurised cabin manufactured since have rounded windows.

On the organisational level things are similar. One particular accident, the Grand Canyon Collision in 1956 (NTSB, 1957) laid the foundations of

²Rumour has it that Sir Winston Churchill himself was personally involved in the enquiry following the crashes.

commercial aviation as we know it today. That particular accident, consisting in a mid-air collision of two passenger airliners over the Grand Canyon, did not involve any technical failure. The two airplanes were in perfect working order. The causes were to be found in the very way flying was (or rather wasn't) organised at the time. Once outside of the immediate vicinities of the airport, the responsibility for maintaining separation and avoiding collisions fell solely on the flight crews. They were tasked with communicating their positions among each other and negotiating with one another to ensure that they pass at a safe distance. In case the radio communications failed, the only barrier preventing collisions was the eyes of the pilots on the lookout for conflicting traffic and the relative vastness of the skies. It was not before long that two planes collided over the Grand Canyon. After the collision, public outcry put enough pressure on government that flight safety became an issue at the very highest political level. A monumental effort was undertaken to ensure that such accidents do not occur in the future in the US leading, among other things to the introduction of continuous radar tracking of flights, minimum separation standards, mandatory flight corridors and the creation of the FAA³, the US state regulator.

A major accident even kicked off voluntary incident reporting. The investigation of a crash in 1974, when a passenger jet flew into a mountain, found out that the crew had misunderstood instructions from ATC⁴ (NTSB, 1975). It also revealed that only six weeks prior to the accident, at the same location, another aircraft had misunderstood the clearance and only narrowly avoided the mountain. The airline had rushed to inform its own flight crews about the danger but, due to a lack of an adequate feedback channel, other airlines had not received any warning. The obvious avoidability of the accident led to an agreement between the FAA and NASA⁵ in 1976 to create and operate a voluntary confidential non-punitive reporting program called ASRS⁶ (§2.1.4). ASRS is currently considered as one of the success stories in voluntary incident reporting and the model is being copied to other industries (Barach and Small, 2000).

2.1 Producing occurrence data

In this section we will provide overview of a representative cross section of the different types of occurrence data that is commonly produced and consumed by the different entities, part of civil aviation's ecosystem. As we saw in the previous chapter (§1.1.3), there is a whole spectrum of reportable occurrences. Putting aside the question of under-reporting (Winder and Michaelis, 2005)

³Federal Aviation Administration

⁴Air Traffic Control

⁵National Aeronautics and Space Administration

⁶Aviation Safety Reporting System

every such event generates information which is stored in an electronic format: **occurrence data**.

2.1.1 The actors

Accident causation (and prevention) can be looked upon as a problem of maintaining control over a system at different hierarchical levels (§1.2). We saw how control is ultimately a function of making adequate and informed decisions based on reliable feedback information. Before going on and speaking about the different types of data produced let us take a look at the actors involved in the civil aviation landscape.

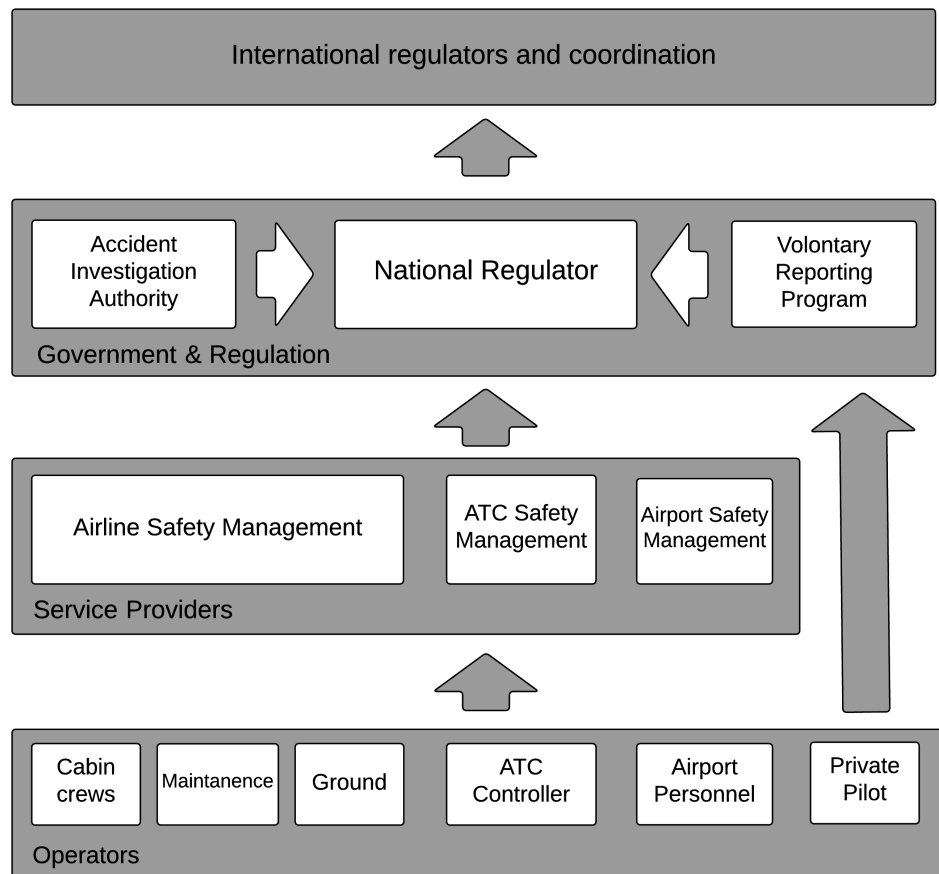


Figure 2.1: Types of actors in civil aviation

Figure 2.1 shows the main types of entities that participate in the system, which we have arranged according to their proximity to the actual physical

processes involved with flying airplanes. At the very bottom are the individuals doing the work (**the operators**): Pilots, technicians, ground crews, air traffic controllers, airport staff etc. . . . Safetywise, they are tasked with controlling the physical processes that they are responsible for: flying, controlling maintaining. The arrows represent the flow of feedback information from the level of operations to the higher levels of the system.

Operators are almost all part of a larger entity (**the service providers**): Airlines, airports, ATC, etc. . . They are responsible for providing and ensuring a safe working environment for their staff by crafting procedures and rules and enforcing them within the relative perimeter.

On the government level, several distinct entities are involved in the safety process. These are the **national regulators**, who are responsible for crafting the rules that each service provider must oblige with as well as for enforcing these rules. In France the DGAC is the national regulator, in the United states it's the FAA, in Canada it's the Ministry of Transport (Transport Canada). At the national level we also find the **accident investigation authorities**, like for example the BEA (France), the NTSB⁷ (USA), the TSB⁸ (Canada) as well as various programs designed for information exchange, such as **voluntary reporting programs**.

Finally, as commercial aviation is not confined to within national borders there are a number of entities that regulate and coordinate the activity on an international level. The most notable are ICAO, and in Europe the EASA and Eurocontrol, responsible for European ATC.

Generally speaking the entities closer to the bottom are the ones that mostly produce feedback data and those closer to the top are the ones that mostly consume it. Also, as data propagates from the bottom up, the higher the entity, the more diverse data sources and data types it accumulates. The DGAC for example collects data from all the service providers as well as from the accident investigation authority and from other regulatory authorities through data exchange programs. Programs such as ASRS effectively bypass the company level and aim specifically at collecting information from the operators on the government level.

However information also propagates from top to bottom. This is called *dissemination*. Accident investigation authorities publish their findings and the information is consumed at lower levels, public data sources (in North America) are maintained by the FAA and the Canadian authorities publishing large amounts of data and, of course, everybody can read the specialised press dedicated to incidents and accidents in aviation.

Besides the above-mentioned entities, there are a number of other actors that produce and consume the data that we are concerned with. These are:

⁷National Transportation Safety Board

⁸Transportation Safety Board of Canada

- Companies such as ASCEND⁹ sell feeds of compiled data about incidents and accidents.
- Insurance companies also keep records of descriptions of incidents.
- Specialised press, such as *The Aviation Herald* (§2.1.6) and websites such as the *Aviation Safety Network* run parallel inquiries and often collate important detail about recent accidents well before the official accident report is published.

Finally worth mentioning is the *Aviation Safety Network*'s "Wikibase" where individuals are encouraged to contribute and maintain a repository of occurrence information on a voluntary basis, effectively crowdsourcing incident data.

We will now see examples of the data produced by these entities.

2.1.2 Official accident investigations

2.1.2.1 The process

As we saw at the beginning of this chapter, major accidents and the subsequent investigations are the main force driving the improvement of safety. The documents produced and published after an official investigation today span decades of operations, concern tens of thousands of accidents and incidents and form a global record.

The authorities carrying out the investigations are specific to each country and rules specify who will conduct the investigation. Usually it is the authority of the country where the accident occurred with the help of experts from the country where the unfortunate aircraft was registered and the country where the aircraft was produced.

The goals of accident investigation are simple:

- Gather the necessary evidence and determine the exact circumstances of the accident.
- Identify the probable causes of the accident as well as any notable factors that contributed to the particular outcome.
- Come up with the measures that need to be taken so that the same event never reproduces itself in the future.

Investigations can take anywhere from a few weeks to several years before all the relevant data is gathered and analysed. The end result is an accident report and often a change in regulation and/or policy aiming at introducing barriers to the specific accident scenario being investigated.

⁹<http://www.ascendworldwide.com/>

2.1.2.2 Report examples

An official accident investigation report details all¹⁰ the relevant facts about the incident, provides information about the investigation itself, exposes the findings of the investigation, determines the probable causes of the accident and produces recommendations about how to improve the system.

Typically most accident reports contain 4 major parts:

- **Factual part:** The facts and circumstances of the accident are presented. Typically this part includes a detailed description of the event unfolding and is a collection of factual data about the event. It is often supplemented by a more thorough description of those aspects of the event or of the circumstances that are relevant to the particular incident. If, for example, weather was a factor the weather conditions will be exposed in detail.
- **Analytical part:** The part which presents the analysis of the investigators based on the gathered facts. Based on those facts the sequence of events that led to the accident is reconstructed and the causes determined.
- **Conclusions and probable cause:** The part which synthetically presents the investigators' conclusions as to what the causes of the accident were.
- **Recommendations:** The actions that should be taken and by whom so that a similar accident does not reoccur in the future.

Figure 2.2 represents the table of contents of the NTSB's report on the Asiana flight 214 accident on July 06 2013 and shows the different sections.

¹⁰Considered relevant by the investigating body

1. Factual Information	1
1.1 History of Flight	1
1.2 Injuries to Persons	13
1.3 Damage to Airplane	13
1.4 Other Damage	14
1.5 Personnel Information	14
1.6 Airplane Information	19
1.7 Meteorological Information	30
1.8 Aids to Navigation	30
1.9 Communications	30
1.10 Airport Information	30
1.11 Flight Recorders	33
1.12 Wreckage and Impact Information	34
1.13 Medical and Pathological Information	35
1.14 Fire	36
1.15 Survival Aspects	37
1.16 Tests and Research	56
1.17 Organizational and Management Information	61
1.18 Additional Information	71
2. Analysis	77
2.1 General	77
2.2 Accident Sequence	78
2.3 Flight Crew Performance	85
2.4 Autoflight System Training and Design	93
2.5 Pilot Training	98
2.6 Operations Issues	101
2.7 Low Energy Alert	104
2.8 Survival Aspects	107
3. Conclusions	126
3.1. Findings	126
3.2 Probable Cause	129
4. Recommendations	130

Figure 2.2: Table of contents of NTSB/AAR-14/01

The different parts have different rhetorical functions. Figure 2.3 shows excerpts from the beginning of the document. There is an overall summary of the accident as well as a very detailed chronological narrative of the accidental sequence. These parts “paint” the overall picture and context and present the facts as they were collected by the investigation authorities.

On July 6, 2013, about 1128 Pacific daylight time,¹ a Boeing 777-200ER, Korean registration HL7742, operating as Asiana Airlines flight 214, was on approach to runway 28L when it struck a seawall at San Francisco International Airport (SFO), San Francisco, California. Three of the 291 passengers were fatally injured; 40 passengers, 8 of the 12 flight attendants, and 1 of the 4 flight crewmembers received serious injuries. The other 248 passengers, 4 flight attendants, and 3 flight crewmembers received minor injuries or were not injured. The airplane was destroyed by impact forces and a postcrash fire. Flight 214 was a regularly scheduled international passenger flight from Incheon International Airport (ICN), Seoul, Korea, operating under the provisions of 14 Code of Federal Regulations (CFR) Part 129. Visual meteorological conditions (VMC) prevailed, and an instrument flight rules (IFR) flight plan was filed.

At 1127:32.3, an electronic voice announced “two hundred.” At 1127:33.6, the PM stated, “it’s low,” and the PF replied, “yeah.” At 1127:36.0, one of the flight crewmembers made an unintelligible comment. At 1127:39.3, the quadruple chime master caution alert sounded. When the alert sounded, the airplane was about 0.45 nm from the runway at 124 ft RA, the airspeed was about 114 knots, and the descent rate was about 600 fpm. At 1127:41.6, an electronic voice announced “one hundred.” At 1127:42.8, the PM stated, “speed.” Less than a second later, both thrust levers were advanced by the PM.²⁴ At 1127:44.7, the A/T mode changed from HOLD to THR. At 1127:46.4, the CVR recorded the stick shaker activating, and the lowest airspeed during the approach of about 103 knots was recorded by the FDR at 1127:46.9. At this time, the airplane was about 0.35 nm from the runway at 39 ft RA, the descent rate was about 700 fpm, the N1 speeds for both engines were increasing through about 50%, and the pitch attitude reached about 12° nose up. The airspeed then began to increase. At 1127:47.8, the PM called out, “go around,” and at 1127:48.6, the airspeed was about 105 knots, and the stick shaker stopped. The initial impact with the seawall occurred at 1127:50. At that time, the N1 speeds for both engines were increasing through about 92%, and the airspeed was about 106 knots.²⁵

Figure 2.3: Excerpts from the “Factual Information - History of Flight” (§1.1, p. 19) section of NTSB/AAR-14/01

For high profile cases, such as the Asiana crash, the level of detail of the facts can be very minute. Figure 2.4 is an excerpt from the “Personnel Information” section where the morning activities of the pilot plying (PF) are presented. Similar sections exist for all three members of the flight crew.

On Saturday, July 6, the PF woke about 0700 feeling rested. He went jogging, returned about 0800, and ate breakfast. He took a bus to ICN about 0930, arrived about 1030, and began preparing for the flight. The official show time was 1510, but he met his instructor (the PM) about 1440, and they began briefing for the flight. The PF had a cup of coffee when he arrived at the airplane.

Figure 2.4: Excerpts from the “Factual Information - Personnel Information” (§1.5, p. 33) section of NTSB/AAR-14/01

The analytical section presents the analysis of the investigation authorities. Its rhetorical function is to argument and present as clearly as possible how the investigators came to their conclusions. Figure 2.5 is an excerpt from this section. We can see how the language changes slightly. Phrasings such as “might be attributable, at least in part, to fatigue” denote that the investigators are presenting their expert opinion rather than stating facts.

The PF made several errors that might be attributable, at least in part, to fatigue. These errors included his selection of FLCH SPD at 1,550 ft without remembering that he had already selected the go-around altitude in the MCP altitude window less than 1 minute earlier, being slow to understand and respond to the observer’s sink rate callouts, not noticing the decrease in airspeed between 500 and 200 ft, and not promptly initiating a go-around after he detected the low airspeed condition.

The PM also made several errors that might be attributable, at least in part, to fatigue. These errors included not noticing the PF’s activation of FLCH SPD at 1,550 ft or subsequent indications on the FMA, not ensuring that a “stabilized” callout was made at 500 ft, not noticing the decay in airspeed between 500 and 200 ft, and not immediately ensuring a timely correction to thrust was made when he detected the low airspeed.

Figure 2.5: Excerpts from the “Analysis - Flight Crew Performance” (§2.5, p. 86) section of NTSB/AAR-14/01

Figure 2.6 shows a passage from the analytical section where a system is described in detail. Their primary function is to define the objects that are discussed in the report.

The 777 was equipped with a low airspeed alerting system that was first certified on the 747-400 in 1996 and then certified for the 777-200B in 1997. This system, developed as a result of a safety-related incident reported by a customer airline in 1995,101 was designed to alert flight crews of decreasing airspeed to avoid imminent stalls. The system, which activates when airspeed decreases 30% into the amber band, was not designed to alert crews that their airspeed had fallen below V_{ref} during approach. According to Boeing, the triggering threshold for the low airspeed alert was selected to avoid nuisance alerts during normal operations and to minimize them during intentional operations at low airspeeds. Minimizing nuisance alerts is an important consideration in the design of alerts because too many false alerts can increase flight crew response times or cause crews to ignore alerts altogether.

Figure 2.6: Description of the “low airspeed alerting system” system (§2.7, p. 104) of NTSB/AAR-14/01

Figure 2.7 is the *probable cause* statement of the investigative authority. In a way this is the official conclusion as to what caused the accident and the concentration of all the work in the report. Generally speaking after a probable cause statement is issued (and the report published) the case is considered closed.

The National Transportation Safety Board determines that the probable cause of this accident was the flight crew’s mismanagement of the airplane’s descent during the visual approach, the pilot flying’s unintended deactivation of automatic airspeed control, the flight crew’s inadequate monitoring of airspeed, and the flight crew’s delayed execution of a go-around after they became aware that the airplane was below acceptable glidepath and airspeed tolerances. Contributing to the accident were (1) the complexities of the autothrottle and autopilot flight director systems that were inadequately described in Boeing’s documentation and Asiana’s pilot training, which increased the likelihood of mode error; (2) the flight crew’s nonstandard communication and coordination regarding the use of the autothrottle and autopilot flight director systems; (3) the pilot flying’s inadequate training on the planning and executing of visual approaches; (4) the pilot monitoring/instructor pilot’s inadequate supervision of the pilot flying; and (5) flight crew fatigue, which likely degraded their performance.

Figure 2.7: Probable Cause (§3.2, p. 147) of NTSB/AAR-14/01

Finally figure 2.8 shows the recommendations section of the report. This is a manifestation of the overall safety process. After the accident, the investigators invite the regulator (in this case the FAA) to craft new legislation as well as different protagonists to reconsider certain aspects of their operations, all in the object off never repeating the same accident.

As a result of this investigation, the National Transportation Safety Board makes the following new safety recommendations:

To the Federal Aviation Administration: Require Boeing to develop enhanced 777 training that will improve flight crew understanding of autothrottle modes and automatic activation system logic through improved documentation, courseware, and instructor training. (A-14-37)

[14 more...]

To Asiana Airlines: Reinforce, through your pilot training programs, flight crew adherence to standard operating procedures involving making inputs to the operation of autoflight system controls on the Boeing 777 mode control panel and the performance of related callouts. (A-14-52)

[3 more...]

To Boeing: Revise the Boeing 777 Flight Crew Operating Manual to include a specific statement that when the autopilot is off and both flight director switches are turned off, the autothrottle mode goes to speed (SPD) mode and maintains the mode control panel-selected speed. (A-14-56)

[1 more...]

To the Aircraft Rescue and Firefighting Working Group: Work with the Federal Aviation Administration and equipment manufacturers to develop and distribute more specific policies and guidance about when, how, and where to use the high-reach extendable turret's unique capabilities. (A-14-58)

[4 more...]

To the City and County of San Francisco: Routinely integrate the use of all San Francisco Fire Department medical and firefighting vehicles in future disaster drills and preparatory exercises. (A-14-62)

[1 more...]

Figure 2.8: Some of the recommendations (§4, p. 148) of NTSB/AAR-14/01

All in all, official accident reports are the most comprehensive source of information about a particular occurrence, and taken as a whole constitute the repository of all that we have learned about why airplanes crash, during almost a century of flying them.

Data-wise these documents present major challenges. They are intended for “human consumption” only and are formatted accordingly. Mostly published as *pdf* files they are difficult to exploit automatically. Even “simple” tasks such as indexing in a *full-text* search engine (§2.3.1) require specific pre-processing to gain access to text in machine-readable form. Furthermore the quantity of (redundant) information they contain and the internal structuring and formatting make it rather difficult to access the relevant parts.

Searching for reports where crew fatigue was a factor, for example will be difficult using off-the-shelf search engines. Querying for the term fatigue will bring far too many reports where the term “fatigue” itself is present without being relevant to the task at hand.

Working with such reports, Thibert (2014) in his master’s thesis demon-

strated that even for (apparently) simple tasks such as the one mentioned above keyword-based approaches are insufficient and factors such as lexical ambiguity and negation need to be taken into account.

Internal structuring and formatting of such documents imply that different sections have different rhetorical roles (Teufel and Moens, 2002). Some are *explanatory*, some are *argumentative*, some are *descriptive*. Automatically discerning the roles of a section has the potential to vastly improve performance of information retrieval (§3.1) and text mining systems by targeting the analysis on those parts of the document that are susceptible to contain the relevant information, rather than on the whole document. If one is interested in extracting the sequence of events, for example one would target the (descriptive) “history of flight” section of a report. Exploring this possibility Campello Rodrigues (2013) showed in his master’s thesis that automatically discerning rhetorical function of the different parts is a feasible task.

We also used official accident reports as a resource for calculating similarity between documents written in two distinct languages. In this system we leveraged the redundancy of the present information and the fact that, in Canada, accident reports are systematically published both in English and in French, to provide a interlingual layer of processing for language independent similarity calculation and achieved encouraging results (§6.3) (Tulechki and Tanguy, 2013)

2.1.2.3 Acquiring the data

Official accident reports are part of the public record. They can be consulted on the websites of the accident investigation authority usually in *pdf* format. More specifically:

- The **BEA** (France) have 2442 reports published on their website¹¹. They are in the form of *pdf* files.
- The **BST** (Canada) have 1093 reports published on their website¹². The reports are available both as *pdf* files and *html* pages. The reports are available in English and in French.
- The **NTSB** (USA) have 463 reports currently published on their website¹³. The full reports are available in pdf format and the summaries as *html* pages. The NTSB also maintain a data exchange program where accident and incident report data sets are available for bulk download and as data feeds in a variety of formats (xml, *ms-access* databases, plain text, ECCAIRS (§2.2.3.5) etc...). The service is described on a dedicated website¹⁴. A total of 76631 records are available through this

¹¹<http://www.bea.aero/>

¹²<http://www.bst-tsb.gc.ca/eng/rapports-reports/aviation/index.asp>

¹³<http://www.ntsb.gov/investigations/AccidentReports/Pages/aviation.aspx>

¹⁴http://www.ntsb.gov/_layouts/ntsb.aviation/index.aspx

service.

2.1.3 Preliminary reports and accidents briefs

2.1.3.1 The process

At the same time as official accident investigations, information about minor incidents is often published by the regulatory or accident investigation authorities. These publications can have different forms and serve several functions. High profile accidents generate considerable public interest and updates are published by the authorities as the investigation progresses. These reports are in the form of preliminary information. Generally they only state a small amount of facts, such as the date, a summary of the incident and make and model type. Figure 2.9 represents such a report published by the Canadian authorities using the CADORS¹⁵ system.

Making such information publicly available is a decision of the authorities in question. While most regulators maintain databases of a large number of incidents, some may decide not to publish them. CADORS is today the most advanced system of publicly available accident briefs. Updated daily it contains over 200,000 reports. The FAA's AIDS¹⁶ is a similar initiative in the United States. In France, the DGAC does not maintain a similar system.

2.1.3.2 Report examples

The report in figure 2.9 represents the narratives from a preliminary report in the Canadian CADORS database. CADORS is unique as they publish information systematically in English and in French. In this report, we can see how the information is progressively updated. The initial notification was published on April 17 and an update giving some more information was added to the record on April 22.

This collection is interesting in part due to the fact that documents are systematically published in two languages (as are the reports from the Canadian TSB), thus making them perfect candidates for building parallel corpora (Véronis, 2000). We used this collection for evaluating the performance of a system for detecting similarities across languages (§6.3).

¹⁵Civil Aviation Daily Occurrence Reporting System

¹⁶Accident and Incident Data System

[2015-04-17] A 2115828 Ontario Inc. Cessna 172N (C-GZTJ) from Vancouver / Boundary Bay, BC (CZBB) to Qualicum Beach, BC (CAT4) experienced an engine failure while doing a VFR photo survey work over Texada Island. The aircraft overturned while landing on a field. Two other aircraft working with C-GZTJ circled overhead awaiting emergency personnel. Pilot was able to walk to a farmhouse with minor injuries.

[2015-04-22] UPDATE: JRCC SARSUM Report[V2015-00599]: (494020N 1242520W - Texada Island). Comox Tower called to report a Cessna C-172, C-GZTJ, had declared a Mayday and was attempting a forced landing on Texada Island after losing engine power. Two companion aircraft circled overhead and relayed the crash position. R904, Cape Kuper and Cape Cockburn were tasked. A resident called to say they had the pilot with them and that the pilot had very minor injuries. The pilot was tended to by Emergency Health Services (EHS), Fire and police and the Corm and Coast Guard vessels were stood down. Transport Canada was notified.

[2015-04-17] Un Cessna 172N de 2115828 Ontario Inc. (C-GZTJ) en provenance de Vancouver / Boundary Bay, C.-B. (CZBB) et à destination de Qualicum Beach, C.-B. (CAT4) a subi une panne de moteur lors d'un levé photographique en vol VFR au-dessus de l'île Texada. L'aéronef a capoté en atterrissant dans un champ. Deux autres aéronefs qui travaillaient avec C-GZTJ tournoyaient en survol en attendant le personnel d'urgence. Le pilote a pu se rendre à pied dans une ferme avec des blessures mineures.

[2015-04-22] MISE À JOUR : Rapport SARSUM [V2015-00599] du JRCC : (494020N 1242520W - île Texada). La tour de Comox a appelé pour signaler qu'un Cessna C-172 (C-GZTJ) avait lancé un appel « Mayday » et essayait d'effectuer un atterrissage forcé sur l'île Texada après avoir subi une perte de puissance moteur. Deux aéronefs qui l'accompagnaient ont décrit des cercles au-dessus du lieu de l'accident et ont transmis la position d'écrasement. R904, Cape Kuper et Cape Cockburn ont reçu la mission. Un résident a appelé pour signaler qu'il se trouvait en présence du pilote et que ce dernier avait subi des blessures très mineures. Les services d'urgences de santé (SUS), les services d'incendie et les services de police se sont occupés du pilote, tandis que le Cormorant et les navires de la Garde côtière ont été libérés. L'incident a été signalé à Transports Canada.

Figure 2.9: Narratives form CADORS accident briefs (nu 2015P0532)

The report in figure 2.10 is an accident brief from the FAA's AIDS database. It is a summary of an accident presented in a concise manner. It is interesting to note the concise writing style and the (deliberate?) choice to use all capital letters, even though this particular occurrence dates from February 2015.

While it might seem trivial, the fact that the text is written in all capital letters might pose a problem for tools that automatically identify sentence boundaries (tokenisers §4.1.1) as they often rely on capitalisation as a cue to determine where a sentence starts (Kiss and Strunk, 2006). The equivalent

in German would also pose a problem for POS-tagging as capitalisation in German indicates a noun.

AIRCRAFT N9602Q DEPARTED RST AIRPORT ENROUTE TO MNM. CLIMB TO 9000 FT, 30-40 MINUTES LATER CLIMB TO 10000 NO ICE. 30 - 40 MILES FROM MNM CONTROLLER TOLD THE PILOT TO DESCEND AT PILOT DISCRETION, PILOT ASK IF THERE WAS ANY ICE REPORTED, CONTROLLER RESPONDED NO ONE FLEW THE ROUTE TO REPORT.[...] LANDED ON RUNWAY 11 HARD AND THE AIRCRAFT SKIDDED TO THE LEFT INTO A SNOW BANK. THE THREE LANDING GEARS BROKE FROM THE AIRPLANE, PROPELLERS HIT THE GROUND AND BENDED. AIRCRAFT FINALLY STOP ABOUT 50 YARDS FROM THE INTERSECTION OF RUNWAY 11 AND THE TAXI WAY. 5 PASSENGER ON BOARD 3 MINOR INJURIES, NO POST CRASH FIRE.

Figure 2.10: FAA AIDS Report nu 20140213000969I

2.1.3.3 Acquiring the data

CADORS has 200933 published reports on its website¹⁷. The reports are in the form of highly structured *html* pages (see fig. 2.20 for a screenshot). Since April 2014, CADORS also provides a data feed by email where two *xml* files (for English and French) are sent on a daily basis.

AIDS data is available on their website¹⁸ and 98865 reports are retrievable through the web service.

2.1.4 Voluntary reporting programs

2.1.4.1 The process

Voluntary reporting programs are provided by regulators aimed at gathering and aggregating information directly from operators (pilots, ATC, airport staff, maintenance, cabin crews, etc...) on perceived dangerous situations. Protected by guarantees like anonymity and often incentivised by non-punitive policies, operators are encouraged to share any deviations they have encountered.

ASRS is the first and without doubt the most famous voluntary incident reporting program. Operational since 1976, it has processed over a million incident reports and averages 6736 monthly (322 daily) submissions (NASA, 2014). Figure 2.11 presents the evolution of submitted incident reports since 1981 and shows how the tendency is on the rise.

¹⁷<http://wwwapps.tc.gc.ca/Saf-Sec-Sur/2/cadors-screaq/m.aspx>

¹⁸<http://www.asias.faa.gov/>

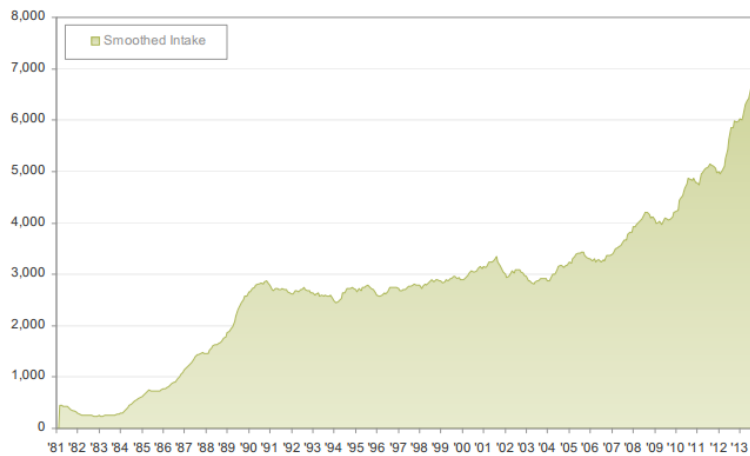


Figure 2.11: ASRS monthly report intake

The basic function of such programs is to collect information about occurrences that are not subject to mandatory reporting (§2.1.5). As is explicitly stated in the ASRS procedure, one should not report events (which should be reported to the FAA through other channels) but only minor incidents and perceived dangerous situations. ASRS thus fills in a “void” effectively descending even further down the failure type pyramid (§1.1.3). ASRS’s is nevertheless a fully functional incident reporting system as it provides feedback loops and independent investigation on reported occurrences. Before they are published on the website, the reports go through an initial screening and if a dangerous situation is identified, ASRS notifies the FAA to take appropriate actions. Also ASRS staff may follow up with the reporters for additional information if such would benefit the safety of the system.

2.1.4.2 Report examples

The report in figure 2.12 is an example of an ASRS report illustrating the high-level organisation of the narrative information. The event involves an aircraft entering onto a runway on which another aircraft is about to take off. The report consists in total of six¹⁹ narratives each written by a different protagonist in the event. Three are written by flight crews and three by ATC controllers working in the tower at the moment. A short synopsis written by ASRS staff summarises the situation.

¹⁹we selected three for illustrative purposes.

Synopsis

Three pilots and three controllers reported an incident where, due to Controller coordination issue, one air carrier started takeoff roll when another was crossing the runway downfield, resulting in an aborted takeoff.

Narrative: 3

There was an unnecessary distraction in the Tower just prior to the event that could have led to the near collision of the aircraft on the runway. For the greater part of the afternoon we had in trail restrictions for our departures, due to weather around our airspace. During this time, the Supervisor we had in the Tower was letting the system work, and the Tower was quiet and calm. This Supervisor was relieved by another Front Line Manager (FLM). The first thing [the new FLM does is] to call Center, and ask for re-routes for the departures for no reason. Like I said before, the system was working fine; we weren't delaying aircraft that could depart. However, in doing this, the FLM was able to get one aircraft exempt from the 20 mile in trail restriction. This aircraft had already taxied out, and in the Local East Bay. This drew the attention of the Local East Controller, along with the Ground East Controller. If their attention wasn't diverted to this unnecessary coordination, they would have been scanning better and possibly been able to stop this event from happening.

Narrative: 5

On our taxi out to Runway XXL we were instructed to hold short of XXL at Taxiway AAA which we complied with. We were then told to cross XXL left on D full length, aircraft on XXL will be position and hold. I confirmed with my First Officer cleared to cross, when we proceeded to cross I noticed the aircraft was commencing the takeoff roll. I immediately added thrust to expedite across and questioned Ground on the clearance. He initially did not respond and then told us to go to Tower. We noticed the CRJ2 had also aborted the takeoff.

Narrative: 6

We were instructed to line up and wait on XXL. After the preceding aircraft rotated we were cleared for takeoff. We took a few seconds on the runway to check for landing traffic and to finish the Takeoff Checklist. After a brief delay of 4-5 seconds the pilot flying (Captain) pushed the thrust levers forward and I was setting the thrust. Shortly after setting the thrust we both noticed the CRJ7 at least half a plane length across the hold short line and continuing to cross in front of us on Taxiway AAA. The Captain called for the abort and initiated the aborted takeoff. Due to the close proximity of the crossing aircraft, I applied the brakes as well. Soon after we had the aircraft stopped, Captain was making an announcement to the passengers and ATC was communicating with us. After the CRJ7 cleared the runway we were instructed to turn right. It is hard to say how things could have been done differently since the CRJ7 was still on Ground Control and we could not hear ATC clearing them to cross XXL. The day VMC conditions definitely allowed us to easily spot the crossing aircraft.

Figure 2.12: Synopsis and narratives of ASRS ACN1002555

Given that ASRS capture data since the 1970s, the form of the reports

evolved considerably over time. In roughly the first two decades of its existence, the system imposed a particular writing style to the report narratives. Rather than writing in standard English, the reports were keyed in using a semi controlled and standardised language, making heavy use of abbreviations for common aviation terms such as *ACFT* for “aircraft” and *WX* for “weather”. The reports were also written using only capital letters. Figure 2.13 shows an example of this writing style, along with its “translation”.

Such reports present issues mainly due to their domain specific terms. A search engine for example (§3.1) needs to be provided with a list of abbreviations and a specific normalisation layer (§4.1.2) in order to be capable of retrieving documents employing such wording.

FLT (flight) WAS SBND (southbound) ON J-209 AND HAD BEEN CLRED (cleared) TO FL390^a BY A PREVIOUS CTLR (controller). OVER SBY^b VOR^c, CLIMBING THRU (through) FL360, TFC (traffic) WAS CALLED BY ZDC^d NBOUND (northbound) AT FL370 AND 4 MI (military) TFC (traffic) WAS OBSERVED AND CENTER THEN HAD US DSND (descend) TO FL350.

^aFlight Level 39000 feet

^bSalisbury–Ocean City–Wicomico Regional Airport

^cVHF Omni Directional Radio Range navigation system

^dWashington Air Route Traffic Control Center

Figure 2.13: Narrative of ASRS ASN45677 using the old writing style

2.1.4.3 Acquiring the data

ASRS data is available online on the website²⁰ and can be searched through a complex search engine. The results of the query can be viewed in *html* format and downloaded in *xls (ms-excel)*, *csv* and *doc (ms-word)* formats. The queries are limited to 5000 results. The whole database is also available upon request through a form on the website and a CD-ROM is sent free of charge containing a *Oracle* database dump of all the data. Currently there are over 160000 reports in the database.

2.1.5 Safety management systems and mandatory reporting

2.1.5.1 The process

A safety management system (or SMS) is a “systematic approach to managing safety, including the necessary organisational structures, accountabilities, policies and procedures” (ICAO, 2001) and they are becoming mandatory all over the world.

²⁰<http://asrs.arc.nasa.gov/>

Incident reporting is a crucial component of any SMS. Companies are required to provide the necessary feedback channels for reporting incidents and ensure that the reports are as truthful as possible, via non-punishment and anonymisation policies.

Such internal incident reporting programs generate potentially large amounts of data. In a large national airline company, for example, production of reports can be over 1000 documents per month.

Furthermore, mandatory sharing regulations oblige operators to forward reports gathered through SMS to the national regulators. In France the DGAC thus receives reports from all the service providers operating on French soil, which amounts to between 4000 and 5000 new reports per month.

2.1.5.2 Report examples

Figure 2.14 shows the narratives of three reports from a large airline company's SMS system. For clarity we chose reports in English, but the database we used contains a mix of both English and French. We can see the uncontrolled writing style, non standard use of punctuation and first person wording.

Another particularity is that the titles are systematically in French. This helps retrieval and access to the reports.

The wording and (lack of) grammar employed in these reports makes all but the most basic language processing very inefficient. Lack or non-standard use of punctuation (such as the use of a semi-colon in place of a period for separating sentences) makes sentence splitting difficult. Token identification can also be tricky, given that terms such as “v/s” contain delimiter characters. The third report is difficult even at the most fundamental level - that of identifying the language in which it is written, (vital information for all language processors). In the same report we can also see a (not so uncommon) encoding issue. The apostrophe is replaced by a question mark, probably during data migration.

REFUS DE REPONSE DE L'ATC A LA DEMANDE DE ROULAGE.

Ready to taxi to RWY 03 at PHC. No answer of the ATC controller after many calls on the tower frequency. It appears that probably the ATC controllers were watching match Nigeria vs South Korea. We have been waiting for 20 minutes before we deserve an answer of the tower controller. What could happen in case of an emergency in the meantime ? It is simply unbelievable !

AIRPROX DEPOSE. MANOEUVRE D EVITEMENT SUITE RA TCAS

AIRPROX On heading 270;descending to fl 100. Atc said:heading 330;then atc said:stop descent now then we had a Tcas Ra:Climb(red until v/s +1000ft/min) estimated separation:800ft.

ANOMALIE FICHE DE TERRAIN

CONSIGNE « ILS 17L GLIDE UNRECIABLE ?? PEU CLAIRE EN DOC AF Chef Jeppesen; un paragraphe « other information » indique : « caution : on approach to RWY 17L/35R expect momentary distorsions or interruptions of GS signal. GS fluctuations depending on taxiing of departing ACFT possible. ATC wil clear landing ACFT for ILS approach without GS. Phraseology will be ?cleared for ILS approach; glide path unreliable?. En carto AF; l'info est quasi absente. En effet; la seule référence est dans le paragraphe NOISE ABATEMENT ! On y lit : « ATC may clear LDG aircraft for ILS approach without GP (31R/17L) the phraseology will be cleared for ILS approach; glide path unreliable?. On constate : 1/ que ce qui motive cet usage est absent. 2/ que le paragraphe choisi n'est pas pertinent (abatement). Du coup; un complément (partiel) est mis en RCNI.

Figure 2.14: Examples of reports from a SMS system

2.1.5.3 Acquiring the data

Reports from service provider's SMS is protected private data. It is not publicly available. The formats differ and are usually custom build solutions integrated in the service providers data management system.

2.1.6 Other sources of occurrence data

2.1.6.1 Specialised data providers

Figure 2.15 is an example of a report from the ASCEND data feed. It is roughly equivalent to accident briefs we saw in section 2.1.3. The information

it contains is factual and concise. The reports are collected from a variety²¹ of sources and are coded and checked internally before publishing.

A330, Hard landing, Caracas

During the final stage of an approach to Runway 10 at Caracas the aircraft apparently developed a high sink rate and touched down hard. The landing was completed safely and the aircraft taxied to the gate for normal passenger disembarkation. Following a hard landing inspection the aircraft was released to return to Paris, however, after take-off, the undercarriage apparently would not retract and the flight returned to Caracas. A more detailed inspection found damage to the main undercarriage. The accident happened in daylight (1425L). Weather; wind, variable, visibility 9,000m in drizzle and cloud, broken at 1,300ft. aircraft was operating a flight from Paris Charles de Gaulle.

Figure 2.15: Report from ASCEND data feed

2.1.6.2 Acquiring the data

ASCEND subscription costs 12000GBP/year and reports are then updated on a regular basis via an *xml* feed.

2.1.6.3 Press

*The Aviation Herald*²² is an example of a free press service, specialised in incident and accident data. The Aviation Herald's founder and editor shares²³ that the project started from a personal collection of incident data that he decided to render public. Through the years, his "constant editorial line" and quest to provide factual and unbiased information has gained him the industry's respect.

In 2012 he reported over a million visits per month. We can but speculate that at the time of this writing the number has increased. Also, by providing a rudimentary commenting system, TAH has managed to engage an active community, consisting largely of aviation professionals discussing the more interesting cases while they unfold.

The report in figure 2.16 is an example narrative from the website.

²¹The provider does not wish to provide details.

²²<http://avherald.com/>

²³The interview is available on the website

Incident: Jetblue A320 near Amarillo on Mar 27th 2012, captain incapacitated by panic attack

A Jetblue Airbus A320-200, registration N796JB performing flight B6-191 from New York JFK, NY to Las Vegas, NV (USA), was enroute at FL340 about 55nm north of Amarillo, TX (USA) when the captain suffered a panic attack and behaved entirely incoherent forcing the first officer to seek assistance by cabin crew and passengers to overpower the captain, lock him out of the cockpit and have him restrained in the passenger cabin. Another Jetblue pilot flying as passenger assisted the first officer while diverting to Amarillo for a safe landing about 20 minutes later.

The airline confirmed the flight diverted because of a medical condition with the captain. Another captain travelling as passenger on the flight joined the first officer in the cockpit for the landing and assumed duties as a captain after landing. The ill captain was taken to a local hospital. A replacement aircraft is going to be dispatched to Amarillo to continue the flight. A replacement Airbus A320-200 registration N624JB reached Las Vegas with a delay of 6.5 hours.

Passengers reported the captain had visited the bathroom and when returning to the cockpit basically went nuts and screamed about terrorists and bombs on the aircraft knocking the cockpit door. The first officer locked him out of the cockpit and had him overpowered. It took six people to sit on the captain to restrain him. Another pilot on board went to the cockpit to assist the first officer for the diversion.

On Mar 28th Federal Authorities filed charges against the captain for interfering with flight crew. A court affidavit claims the captain told the first officer they were not going to Las Vegas and started ranting, then left the cockpit and began shouting about a bomb.

The airline reported on Mar 28th the captain has been suspended. He is still in hospital care.

Figure 2.16: Report from The Aviation Herald

2.1.6.4 Community efforts and user generated content

Finally, websites such as *The Aviation Safety Network*²⁴ also provide useful data about incident and accidents. Not affiliated with any official organism, this site is run by a non for profit organisation and aviation enthusiasts who have compiled a comprehensive collection of occurrence reports, constantly updated as new events occur.

Figure 2.17 is a narrative from an ASN report.

²⁴<http://aviation-safety.net/>

A Swearingen SA226-TC Metro II turboprop aircraft was destroyed when it burst into flames after impacting the side of a highway, shortly after takeoff from Querétaro Airport (QRO), Mexico. All five on board suffered fatal injuries, according to authorities. Preliminary reports indicate the aircraft operated on a post-maintenance test flight. Local news sources report that the aircraft went down on the side of Highway 57 Querétaro-Mexico City close to the TransporMex building, some 11 km southwest of the airport.

Figure 2.17: Report from ASN

2.1.6.5 Acquiring the data

Currently there are 16865 articles on the Aviation Herald website²⁵ in html format. On ASN's website²⁶ there are 15800 reports published by ASN staff and 162336 crowd sourced reports in the *wikibase*²⁷. Both are available in *html* format. See figure 2.19 for a screenshot.

2.1.7 A typology of occurrence reports

Now that we saw the main types of information we can summarise their general characteristics. We will provide an external typology, based on the different situational characteristics and an internal typology, listing the main differences in the documents themselves.

2.1.7.1 External categorisation

Table 2.1. Inspired from (Biber, 1993) we distinguish the collections by their situational characteristics:

- **Type:** The type of entity responsible for maintaining the collection (§2.1.1).
- **Occurrence class:** The type of event that generated the report (§1.1).
- **Producer:** The person or institution responsible for writing the report.
- **Purpose:** What the purpose of the document is.
- **Addressee:** Who the document is addressed to.
- **Published:** Whether the document is available to the public.
- **Edited:** If document is subject to an editorial process.
- **Dynamic:** Whether the document is changed or updated over time.

²⁵<http://avherald.com/>

²⁶<http://aviation-safety.net/>

²⁷<http://aviation-safety.net/wikibase/>

Collection	Type	Occurrence class	Producer	Addressee	Purpose	Published	Edited	Dynamic
ASRS	Voluntary Reporting	Incidents	Operator Institution	Aviation community Authorities	Report error Inform about danger	Yes	Corrections Deidentification	No
CADORS	Regulator	Accidents Incidents	Institution	Aviation community	Inform	Yes	<i>NA</i>	Yes
TSB BEA NTSB	Investigators	Accidents	Institution	General public Aviation community Authorities Specific entities	Inform Explain Persuade Rule	Yes	Proofread	No
Aviation Herald	Press	Accidents Incidents	Private initiative	Aviation community General Public	Inform Explain	Yes	Autopublication	Yes
ASN	Press	Accidents	Private initiative	Aviation community General Public	Inform	Yes	Autopublication	Yes
ASN Wikibase	Wiki	Accidents	Community (user generated)	Inform	Aviation Community	Yes	Autopublication	Yes
DGAC	Regulator	Accidents Incidents	Multiple (aggregation)	Regulator Company	Multiple (aggregation)	No	Raw	Yes
ASCEND	Data Provider	Accidents	Institution (commercial)	Businesses Institutions	Inform	Limited	<i>NA</i>	<i>NA</i>
Internal SMS	SMS	Incidents	Operator	Management	Report error Inform about danger Mandatory report Express opinion	No	Raw	No

Table 2.1: External typology of occurrence report corpora

2.1.7.2 Internal categorisation

We will now see the most important aspects of internal variation of the documents. Without doubt the most notable one is the quantity of information present in a report. Figure 2.18 shows the size distribution of documents in four different databases we have worked with, for which we calculated the distributions of documents according to their size. These are:

- **BST:** (390 documents) The Canadian safety investigation authority
- **Aviation Herald (AvH):** (167343 documents) A specialised press service for incident data
- **ASRS:** (13090 documents). US voluntary reporting program
- **DGAC:** (136851 documents) The French regulatory authority
- **SMS:** A large airline's internal reporting program

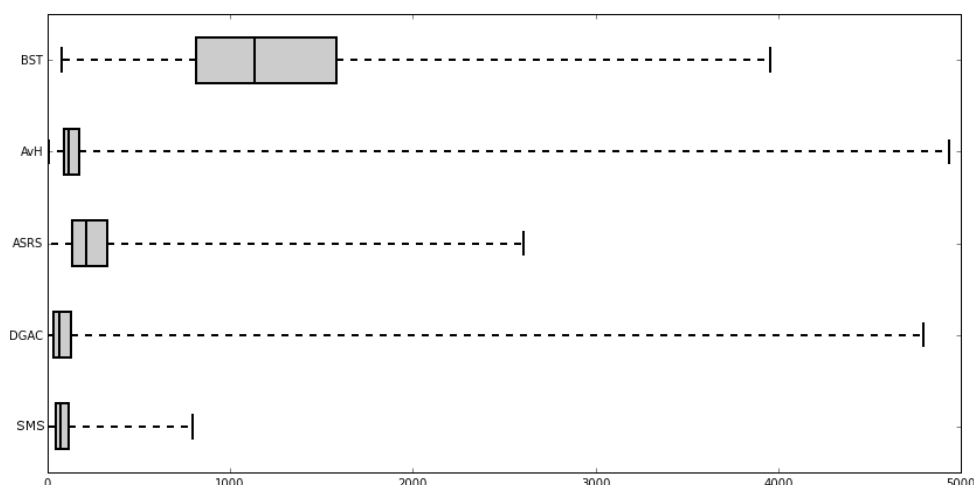


Figure 2.18: Size comparison of reports from various sources in number of words

The boxplots in figure 2.18 show the distribution of document size across the database. The whiskers show the extremes. First of all we can see that, besides the official accident reports of the BST (1248 words on average), documents tend to be relatively short, about 100 words for SMS and DGAC and 200 words for ASRS and AvH on average.

In every collection, there tend to be some much longer documents. Both the DGAC's and The Aviation Herald's databases contain reports of over 4000 words, corresponding to high profile accident investigations for which a lot of information is generated. In both cases this is not surprising. The

DGAC collect occurrences from a variety of sources and thus work both with very succinct incident reports coming from service providers and long accident reports coming from the BEA and other investigative authorities through data exchange channels. The Aviation Herald reports on accidents and depends on the amount of information their staff is able to gather. Sometimes they work with accident briefs and sometimes they report all through the investigation process of a major crash, thus accumulating vast amounts of data.

Reports from the SMS tend to be relatively stable with only a few documents longer than 500 words and so do ASRS's. This stability is due to the fact that these databases are constituted by a single unique process and thus the reports are comparatively homogeneous.

We can also distinguish between the collections according to the following characteristics:

- **Informational content:** Another way reports vary in style is by the type of information that is present. Official accident investigations, as we saw collect and organise all the available information about an occurrence. They are **exhaustive**. This is not always the case for other types of reports. They tend to fall in two categories - **partial** and **brief**. Brief reports are those accounts of incidents that concisely present the event and its circumstances. They are written by independent entities not involved in the event. Examples of such reports are the ones from CADORS (fig.2.9) and the data in the ASN database (fig 2.17). On the other hand we have partial, first person accounts, such as the ones in ASRS (fig. 2.12) and the reports from the SMS (fig. 2.14). They are much more subjective and present the particular point of view of their authors and describe only a selected aspect of the event. Such reports are by definition not exhaustive.
- **Writing style:** Writing style also varies across reports. Official accident reports (§2.1.2.2) are formal documents written carefully and proofread before publication. Other cases, such as first person narratives written (or typed into an i-pad) on the fly on a cockpit table (such as the example in fig. 2.14) exhibit non standard use of punctuation, spelling mistakes and a mix of standard and very technical language. Other cases, such as early reports from ASRS (fig. 2.13) and AIDS reports (fig. 2.10) use all capital letters. Early ASRS reports also use a standardised set of aviation abbreviations, a remnant from the times when screen real estate was a scarce commodity.
- **Structure:** Most reports exhibit little or no structure. Short narratives and rarely even contain paragraphs. Official accident reports are on the other hand *semi-structured* documents, with multiple levels of headings (Feldman and Sanger, 2007). Some of the articles in the Aviation Herald

exhibit **weak structure** as they have zones with different functions, but no explicit signalling such as headings.

- **Multimodality:** Except for metadata, which will be discussed in the next section, reports are mostly monomodal, as they only contain text. Official accident reports and press articles on the other hand also contain diagrams, tables, and pictures. The online sources, ASN and the Aviation Herald also contain video material and hyperlinks to external sources, such as general press articles about incidents or the official *pdf* report.

All in all occurrence data can take many forms and serve various initial functions. One thing is common to all reports - they might carry very valuable information. We will now see the various storage solutions that help organise this data and give access to the information it contains.

Collection	Corpus size (nb docs)	Mean length (nb. words)	Informational content	Writing style	Structure	Multimodality	Language
ASRS	167,000	254	subjective partial	First person abbreviations CAPS	multiple narratives organised by taxonomy	None	English
CADORS	200,933	≈ 150	partial	third person formal	None	None	English French
TSB BEA NTSB	TSB: 1093 BEA: 2432 NTSB: -	1249 (TSB)	exhaustive	formal third person	semi-structured documents	images diagrams tables	English French
Aviation Herald	13,090	214	brief to exhaustive	formal, third person	weakly-structured	images diagrams tables videos hyperlinks	English
ASN	19,127	NA (short accounts)	brief	formal, third person	None	images diagrams tables videos hyperlinks	English
ASN Wikibase	162,336	NA (very short accounts)	brief	formal, third person	None	images diagrams tables videos hyperlinks	English
DGAC	443,181	106	mixed	mixed	None	unstructured	French some English
SMS	NA	92	subjective partial	informal first person use of abbreviations	None	unstructured	French English

Table 2.2: Internal typology of occurrence report corpora

2.2 Storing and organising occurrence data

In the previous section we saw examples from the different types of information that is out there. Now let us take a closer look at how it is stored and managed.

2.2.1 The occurrence and its lifecycle

Using data implies organising it in collections. Whether this data is stored in folders on a library shelf or in complex relational databases one must first define the highest level of discreteness - that of the record. Fortunately, when thinking about accidents and incidents, their discreteness is implied by definition and thus directly transposable to whatever data model is being created.

The term “occurrence” is used to denote a record in an incident database in order to lift any ambiguity with “event” as, at a finer grain, an occurrence can be represented as a sequence of discrete events (see the example in fig. 2.24). An occurrence in a given incident collection can be defined as **the entire body of recorded information related to a specific incident**.

An occurrence is created when news that an incident has taken place arrives at whatever institution or entity is responsible for the collection. In its absolute minimal form an occurrence can be only a date, a location and an indication about the existence of an incident. “A airplane crashed today in Russia” is enough to be considered a valid occurrence, albeit not very informative. At the other end of the spectrum is the official accident report of the same incident, filed months or years after the event, containing all the facts and information considered pertinent by the corresponding investigative body.

As we saw in the previous chapter (§1.1.4), most, if not all reporting and collecting architectures imply some form of “lifecycle” of the occurrence. In some cases, such as internal incident reporting programs the lifecycle is associated with the investigation and treatment process. Additional information is added progressively to the occurrence either by experts working with the initial data or by explicitly seeking it out through the follow-up investigation. In programs where the accent is on collecting and/or aggregating data from multiple sources, occurrences are gradually updated and folded together as novel information about the event is received.

2.2.2 Accident models, coded data and taxonomies

We saw in the previous chapter how thinking about accidents has evolved and is, today both complex and abstract. Modelling efforts, such as the one presented in section 1.2 are becoming ever more complex and the analysis of the causes looking ever further from the actual events. A failing component, for example, may be analysed as a dysfunction of the application of a novel regulation about maintenance procedures. With applied systemic models not

far over the horizon, complexity of the analysis is naturally expected to grow (Hollnagel, 2004).

Just the pure factual data required to understand a modern aviation mishap is of considerable scale. Such facts include, for example information about the context of the occurrence (place, time, meteorological conditions) about the aircraft (make, model, series, maintenance history), information related to human factors (age and experience of the pilot(s), crew composition) and many more.

When considering occurrences on an isolated basis this complexity isn't an immediate hurdle. One can choose a particular analysis methodology, adapted for the occurrence in question. The gathered facts can be organised in an *ad-hoc* manner in, say a spreadsheet. Or as is the case with official accident investigations, just describe the accident and the analysis using natural language.

When designing a system to store and analyse occurrence data, however, all the choices related to the particular organisation of both analytical and factual data have to be done prior to storing the first occurrence. They also have to be generic and applicable to all stored occurrences. Designing the underlying data-model is, thus a major issue. It basically boils down to the question of how much of the occurrence's description (both factual and analytical) is reflected in the data model and usable via standard database queries or retrievable via an indexing scheme.

Even at a relatively low level of granularity, dozens of fields are required in order to accommodate bits of information such as the make and model of the aircraft, the components that failed, the conditions of the flight, where the accident occurred, where the aircraft departed from and was bound for, etc.

At a higher granularity, information such as the service history of the pilot and the first officer, the wetness of the runway, or the quality of response of the emergency rescue services may be required.

Collectively we refer to these attributes as **factual metadata**.

When the database is required to support not only the factual description, but the subsequent analysis of the accident, one has to choose an underlying accident model. As we will see, at one end of the spectrum we can have a collection that simply stores occurrences in a flat fashion, (as files or database records), where all the relevant information is only present in human-readable form, such as a simple narrative or more complex (semi-)structured files, such as official accident reports in *pdf* format (§ 2.1.2). On the other end of the spectrum, specifically designed collections allow access to all relevant bits of information concerning the occurrence. An example of such a system is the ECCAIRS environment (§2.2.3.5) and the ADREP taxonomy, where each potentially relevant bit of information is represented by a dedicated field of a certain type.

This type of metadata we refer to as **analytical metadata**, as it requires expert reasoning or inference in order to produce.

We will now see several examples of coding solutions.

2.2.3 Examples of metadata

2.2.3.1 Simple factual information

The Aviation Safety Network’s website provides systematic coding of several attributes for each report. Figure 2.19 shows a report with its associated metadata. Information about the outcome such as the number of fatalities, the location, and damage to the aircraft is given alongside its make and model name and the make of its engines.

Accident description

Status: Preliminary

Date: Wednesday 18 February 2015

Time: 14:30



Type: [Airbus A300B4-605R\(F\)](#)

Operator: [Unique Air](#)

Registration: A6-JIM

C/n / msn: 643

First flight: 1992-06-09 (22 years 9 months)

Engines: 2 [General Electric CF6-80C2A5](#)

Crew: Fatalities: 0 / Occupants:

Passengers: Fatalities: 0 / Occupants: 0

Total: Fatalities: 0 / Occupants:

Airplane damage: Substantial

Location: Sharjah Airport (SHJ)  [United Arab Emirates](#)

Phase: Standing (STD)

Nature: -

Departure airport: -

Destination airport: -

Narrative:
The Airbus A300B4-605R(F) cargo plane, A6-JIM, was parked on the platform at Sharjah Airport (SHJ) when the nose landing gear retracted, causing the nose to contact the ground. The left hand forward door was open and separated from the fuselage.
The aircraft was undergoing maintenance at the time of the occurrence.

Figure 2.19: Metadata in ASN

2.2.3.2 Standard descriptors of the accident sequence

Similar to the example from the previous section, reports from CADORS also provide metadata. In Figure 2.20 we can see the same sort of factual information as in the ASN report, but also a set of descriptors of the accident in the form of *occurrence categories* and a list of discrete events in the “aircraft events” part. These are coded by CADORS staff upon analysis of the event and represent analytical metadata.

Record # 1

CADORS Number:2015P0532

Occurrence Category(ies):

- Loss of control - ground
- Other
- System/component failure or malfunction [powerplant]

Occurrence Information

Occurrence Type:Accident

Occurrence Date:2015-04-12

Occurrence Time:2125 Z

Day Or Night:day-time

Fatalities:0

Injuries:1

Canadian Aerodrome ID:

Aerodrome Name:

Occurrence Location:Texada Island, BC

TC Region:Pacific Region

Province:British Columbia

World Area:North America

Country:Canada

AOR Number:185855-V1

Reported By:

- NAV CANADA
- Search and Rescue
- Transport Canada

TSB Class Of Investigation:

TSB Occurrence No:

Occurrence Event Information

Aircraft Information

Registration Mark:GZTJ

Foreign Registration:

Flight #:

Flight Rule:VFR

Aircraft Category:Aeroplane

Country of Registration:Canada

Make:CESSNA

Model:172N

Year Built:1977

Amateur Built:No

Engine Make:AVCO LYCOMING

Engine Model:O-320-H2AD

Engine Type:Reciprocating

Gear Type:Land

Phase of Flight:Cruise

Damage:Unknown

Owner:2115828 Ontario Inc.

Operator:2115828 ONTARIO INC. (15476)

Operator Type:Commercial

CARs Subpart:702

Aircraft Event Information

- Engine failure
- Forced landing
- Overturn

Figure 2.20: Occurrence information in CADORS

ACN: 1002555	
Time / Day <i>Date</i> : 201204 <i>Local Time Of Day</i> : 1801-2400	<i>ASRS Report Number.Accession Number</i> : 1003030 <i>Human Factors</i> : Distraction
Place <i>Locale Reference.Airport</i> : ZZZ.Airport <i>State Reference</i> : US <i>Altitude.AGL.Single Value</i> : 0	Person : 4 <i>Reference</i> : 4 <i>Location Of Person.Aircraft</i> : Y <i>Location In Aircraft</i> : Flight Deck <i>Reporter Organization</i> : Air Carrier <i>Function.Flight Crew</i> : First Officer <i>Function.Flight Crew</i> : Pilot Flying <i>Qualification.Flight Crew</i> : Commercial <i>ASRS Report Number.Accession Number</i> : 1003083 <i>Human Factors</i> : Communication Breakdown <i>Communication Breakdown.Party1</i> : Flight Crew <i>Communication Breakdown.Party2</i> : ATC
Environment <i>Flight Conditions</i> : VMC <i>Light</i> : Daylight	Person : 5 <i>Reference</i> : 5 <i>Location Of Person.Aircraft</i> : Y <i>Location In Aircraft</i> : Flight Deck <i>Reporter Organization</i> : Air Carrier <i>Function.Flight Crew</i> : Captain <i>Function.Flight Crew</i> : Pilot Flying <i>Qualification.Flight Crew</i> : Air Transport Pilot (ATP) <i>ASRS Report Number.Accession Number</i> : 1003084 <i>Human Factors</i> : Communication Breakdown <i>Communication Breakdown.Party1</i> : Flight Crew <i>Communication Breakdown.Party2</i> : ATC
Aircraft : 1 <i>Reference</i> : X <i>ATC / Advisory.Tower</i> : ZZZ <i>Aircraft Operator</i> : Air Carrier <i>Make Model Name</i> : Regional Jet 200 ER/LR (CRJ200) <i>Crew Size.Number Of Crew</i> : 2 <i>Operating Under FAR Part</i> : Part 121 <i>Flight Plan</i> : IFR <i>Flight Phase</i> : Takeoff <i>Route In Use</i> : None	Person : 6 <i>Reference</i> : 6 <i>Location Of Person.Aircraft</i> : X <i>Location In Aircraft</i> : Flight Deck <i>Reporter Organization</i> : Air Carrier <i>Function.Flight Crew</i> : First Officer <i>Function.Flight Crew</i> : Pilot Not Flying <i>Qualification.Flight Crew</i> : Commercial <i>ASRS Report Number.Accession Number</i> : 1005043 <i>Human Factors</i> : Communication Breakdown <i>Communication Breakdown.Party1</i> : Flight Crew <i>Communication Breakdown.Party2</i> : ATC
Aircraft : 2 <i>Reference</i> : Y <i>ATC / Advisory.Ground</i> : ZZZ <i>Aircraft Operator</i> : Air Carrier <i>Make Model Name</i> : Regional Jet 700 ER/LR (CRJ700) <i>Crew Size.Number Of Crew</i> : 2 <i>Operating Under FAR Part</i> : Part 121 <i>Flight Plan</i> : IFR <i>Flight Phase</i> : Taxi <i>Route In Use</i> : None	Events <i>Anomaly.ATC Issue</i> : All Types <i>Anomaly.Conflict</i> : Ground Conflict, Less Severe <i>Detector.Automation</i> : Air Traffic Control <i>Detector.Person</i> : Flight Crew <i>When Detected</i> : In-flight <i>Result.Flight Crew</i> : Rejected Takeoff <i>Result.Air Traffic Control</i> : Issued Advisory / Alert <i>Result.Air Traffic Control</i> : Issued New Clearance
Person : 1 <i>Reference</i> : 1 <i>Location Of Person.Facility</i> : ZZZ.Tower <i>Reporter Organization</i> : Government <i>Function.Air Traffic Control</i> : Local <i>Qualification.Air Traffic Control</i> : Fully Certified <i>ASRS Report Number.Accession Number</i> : 1002555 <i>Human Factors</i> : Human-Machine Interface <i>Human Factors</i> : Situational Awareness	Assessments <i>Contributing Factors / Situations</i> : Human Factors <i>Primary Problem</i> : Human Factors
Person : 2 <i>Reference</i> : 2 <i>Location Of Person.Facility</i> : ZZZ.Tower <i>Reporter Organization</i> : Government <i>Function.Air Traffic Control</i> : Supervisor / CIC <i>Qualification.Air Traffic Control</i> : Fully Certified <i>ASRS Report Number.Accession Number</i> : 1002968	
Person : 3 <i>Reference</i> : 3 <i>Location Of Person.Facility</i> : ZZZ.Tower <i>Reporter Organization</i> : Government <i>Function.Air Traffic Control</i> : Other / Unknown	

Figure 2.21: Metadata of ASRS ASN1002555

2.2.3.3 The ASRS coding schema

Alongside the narrative data, ASRS (§2.1.4) provides extensive description of the accident with metadata. These descriptors are coded by ASRS staff upon reception and analysis of the occurrence.

The ASRS taxonomy (ASRS, 2014) is organised around seven high level concepts (or *entities*):

- **Time:** factual information about when the event occurred.
- **Place:** information about where the event occurred.
- **Environment:** information about the context of the occurrence, such as meteorological conditions, lighting and flight conditions.
- **Aircraft:** Information about each implied individual aircraft.
- **Component:** Information about individual (failing) components, such as their manufacturer and the the eventual problem with the component.
- **Person:** Information about the people involved, their certification, role etc. . .
- **Events:** The abnormal events constituting the occurrence.
- **Assessment:** Information about the analysis of the occurrence by ASRS staff. What the primary problem and the contributing factors were.

Each *entity* is specified by a number of (hierarchically) organised attributes with constrained values.

Person for example is characterised by the attributes “*Function*”, “*Qualification*” and “*Experience*”, each having a different set of potential values.

Figure 2.21 shows the metadata of the report in figure 2.12. We can see information about each of the redactors of the six narratives in the *Person* entities, information about the two aircraft involved in the incident in the *Aircraft* entities and general information such as the time of day and the location²⁸ in the *Time / Day*, *Place* and *Environment* entities. Analytical metadata is present both in the form of the *Assessment* entity and in some of the attributes, such as the *Human Factors* attributes of the *Person* entities.

Querying ASRS is possible via a web interface on the ASRS website²⁹. We will discuss in details the query language and the query builder in section 2.3.1.

2.2.3.4 SMS systems and the bow-tie model

SMS systems (§2.1.5) sometimes rely on relatively abstract ways of categorising incidents and use the *bow-tie* or *barrier* model (Reason, 2000).

The Bow-Tie Accident Model (fig. 2.22) represents a synthetic view of an accident scenario, combining both causal and consequential information. It is centred on the concept of hazard, or “unwanted event” (e.g. “Level

²⁸Which in this particular report is anonymised to protect the ATC personnel’s identities.

²⁹<http://asrs.arc.nasa.gov/>

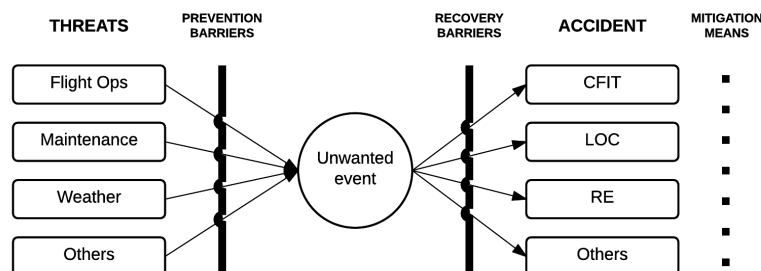


Figure 2.22: The bow-tie accident model

Bust” or “Communication Loss”). Once the hazard is identified, a fault tree is built on the left hand side, representing the cause of the hazard in the form of a set of threats which have contributed to it and a set of barriers which have (or have not) prevented these threats from contributing to the hazard (for example “MTO: Turbulence” or “A/C: Noisy Cockpit”). On the right hand side, an event tree is built representing the barriers that allowed recovery from the hazard, as well as the potential accident and the potential mitigation measures that may or may not have been put in place. Hazards which have not occurred due to proper functioning of prevention barriers are also represented.

Categorising incident reports within this schema requires the coder to choose an item from sets of categories which list all identified threats, barriers, unwanted events, mitigation means and potential accidents (like “CFIT” or “Loss of Control”).

Once categorised, individual reports are exploited both in a quantitative way by producing statistics and trends and in a qualitative way, where the categorisation is queried to identify and extract individual reports of interest for further investigation (§2.3).

Such modelling requires a specific taxonomy organising the various attributes and their relations. It is important to note that such taxonomies could be very abstract and require extensive reasoning from the part of the people responsible for the coding. Imagining the possible (but avoided) outcome means that considerable information is added to the incident record by the coders and is the result of their analysis.

2.2.3.5 ECCAIRS and ADREP

Finally without doubt the most complex system for storing and manipulating incident data is ECCAIRS associated with the ADREP³⁰ taxonomy.

³⁰Accident/Incident Data Reporting

ECCAIRS (European Coordination Centre for Accident and Incident Reporting Systems) is an ongoing effort at standardising accident and incident data collection and exchange within the European Union (Menzel, 2004). Developed by the European Commission’s Joint Research Center, ECCAIRS’s mission is “to assist national and European transport entities in collecting, sharing and analysing their safety information in order to improve public transport safety” and is freely available to any interested party. It takes the form of a software platform that covers most of the collection, indexing and querying of incident reports. Figure 2.23 shows the user interface of ECCAIRS.

Every European country maintains an ECCAIRS database, and these are merged at the community level by EASA³¹. The ECCAIRS software platform allows for complex querying of the databases, with a clear focus on helping the user manage the complexity of the taxonomy, at the expense of textual search. ECCAIRS databases are rarely³² public and their target users are safety managers and analysts. The French DGAC, with whom *Safety Data* collaborate on several projects also uses this software.

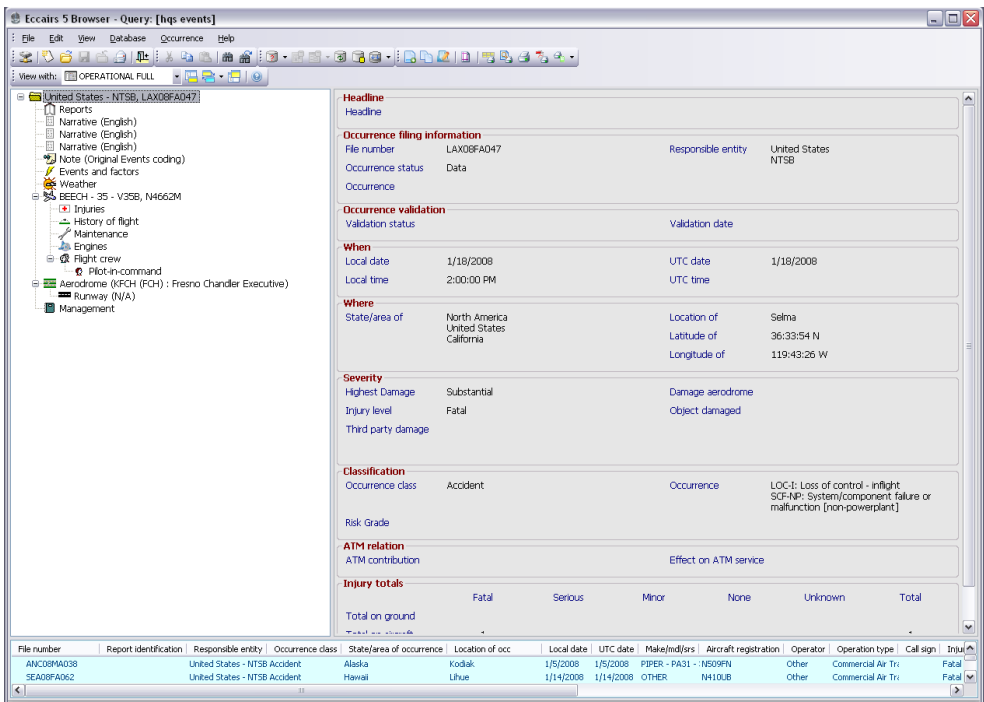


Figure 2.23: ECCAIRS GUI

³¹European Aviation Safety Agency
³²The FAA distribute occurrences in an outdated ECCAIRS format (e4f) via the ASIAs website (<http://www.asias.faa.gov/>).

The ECCAIRS philosophy is to be able to represent a maximum of useful information in a controlled and systemic manner. Unlike official accident reports, that rely mostly on natural language, narrative data in ECCAIRS is just one node of a complex multilevel taxonomy used to represent occurrences called ADREP.

The taxonomy is the result of an effort at standardisation of aviation incident and accident information supported by ICAO (Stephens et al., 2008) and is intended for a very broad coverage. ADREP is an international standard and thus needs to potentially adapt to every possible situation and scenario. Factual descriptors (time, place, aircraft models, engine and component manufacturers etc.) and analytical descriptors of the occurrence, such as *event types* and *explanatory factors* are organised in a complex multilevel hierarchy with more than 800 attributes and 160,000 possible values.

Most interesting are the analytical branches of the taxonomy used to model the accident or incident scenario. The *Occurrence Category* branch provides a high-level description of the corresponding event. In theory, every event can be reliably categorised using one or more of the 36 labels. A consistently labelled database would allow safety experts to examine trends and statistics based on the labels, as well as filtering incident searches by label. Like the rest of the ADREP taxonomy, the labels themselves are normalised and are associated with a set of conditions that describe when they should be used. Table 2.3 shows the list of possible values.

While occurrence categories are relatively simple, ADREP provides a much more detailed way of describing almost any incident by constructing an event sequence using the *Event* entities and combining them with secondary and tertiary attributes from a set of (very large) closed lists of descriptors.

Figure 2.24 represents the accident sequence of the Concorde accident of July 2000 in the form of a sequence of events in ECCAIRS format. To better understand this information we have provided the probable cause statement of the official report in figure 2.25, it gives roughly the same information in narrative form.



Figure 2.24: Sequence of events in ECCAIRS for the Concorde crash

3.2 Probable Causes The accident was due to the following causes:

- High-speed passage of a tyre over a part lost by an aircraft that had taken off five minutes earlier and the destruction of the tyre.
- The ripping out of a large piece of tank in a complex process of transmission of the energy produced by the impact of a piece of tyre at another point on the tank, this transmission associating deformation of the tank skin and the movement of the fuel, with perhaps the contributory effect of other more minor shocks and /or a hydrodynamic pressure surge.
- Ignition of the leaking fuel by an electric arc in the landing gear bay or through contact with the hot parts of the engine with forward propagation of the flame causing a very large fire under the aircraft's wing and severe loss of thrust on engine 2 then engine 1.

In addition, the impossibility of retracting the landing gear probably contributed to the retention and stabilisation of the flame throughout the flight.

Figure 2.25: Probable causes section of BEA report on the Concorde crash

Each event from the sequence is a set of assembled attributes. The fourth

event in the sequence, for example, is composed of four different descriptors:

- The **Event Type**: (“Aircraft wing related event” in blue) comes from a four level hierarchy listing all possible events that can occur on an aircraft.
- The **Flight Phase**: (“during Take-off run” in blue) comes from a separate list and specifies at which point of the flight the particular event occurred.
- A cross reference to the **Aircraft** (“F-BTSC” in blue) specifies the registration of the aircraft concerned by the event (in this case the Concorde)
- The **Descriptive factor** (“Wing plates/skins” in red) further specifies the event by providing the part of the wing that was affected
- The **Explanatory factor** (“Aircraft manufacturing design staff” in black) specifies which elements of the system should be addressed in order to correct the problem.

The ADREP taxonomy has proven to be very useful when used correctly, facilitating data exchange and providing a common frame of reference when speaking about incidents and accidents in aviation (Stephens et al., 2008).

However most of the time, fine-grained categorisation is simply not available, as in the case of the DGAC database we are working with, where only a third of the occurrences are coded with the *occurrence category*, and even less for more precise information such as *event types*, which is the main branch in ADREP for abstracting information about the precise sequence of sub-events that occurred.

	Acronym	Term and detail
Primary categories	ARC	Abnormal runway contact - Any landing or takeoff involving abnormal runway or landing surface contact.
	BIRD	Birdstrike - Occurrences involving collisions / near collisions with bird(s) / wildlife; A collision / near collision with or ingestion of one or several birds.
	CFIT	Controlled flight into or toward terrain - Inflight collision or near collision with terrain, water, or obstacle without indication of loss of control.
	CTOL	Collision with obstacle(s) during take-off and landing - Collision with obstacle(s), during take-off or landing whilst airborne.
	F-NI	Fire/smoke (non-impact) - Fire or smoke in or on the aircraft, in flight or on the ground, which is not the result of impact.
	GCOL	Ground Collision Collision while taxiing to or from a runway in use.
	LOC-I	Loss of control - inflight Loss of aircraft control while or deviation from intended flightpath inflight.
	MAC	Airprox/ ACAS alert/ loss of separation/ (near) midair collisions Airprox, ACAS alerts, loss of separation as well as near collisions or collisions between aircraft in flight.
	RAMP	Ground Handling Occurrences during (or as a result of) ground handling operations.
	RE	Runway excursion A veer off or overrun off the runway surface.
	RI-A	Runway incursion - animal Collision with, risk of collision, or evasive action taken by an aircraft to avoid an animal on a runway or on a helipad/helideck in use.
	RI-VAP	Runway incursion - vehicle, aircraft or person Any occurrence at an aerodrome involving the incorrect presence of an aircraft, vehicle or person on the protected area of a surface designated for the landing and take-off of aircraft.
	SCF-NP	System/component failure or malfunction (non-powerplant) Failure or malfunction of an aircraft system or component - other than the powerplant.
	SCF-PP	Powerplant failure or malfunction Failure or malfunction of an aircraft system or component - related to the powerplant.
	USOS	Undershoot/overshoot A touchdown off the runway surface.
Secondary categories	ATM	ATM/CNS Occurrences involving Air traffic management (ATM) or communications, navigation, or surveillance (CNS) service issues.
	LOC-G	Loss of control - ground Loss of aircraft control while the aircraft is on the ground.
	TURB	Turbulence encounter In-flight turbulence encounter
	FUEL	Fuel related One or more powerplants experienced reduced or no power output due to fuel exhaustion, fuel starvation/mismanagement, fuel contamination/wrong fuel, or carburetor and/or induction icing.
	ADRM	Aerodrome Occurrences involving aerodrome design, service, or functionality issues.
	LALT	Low altitude operations Collision or near collision with obstacles/objects/terrain while intentionally operating near the surface (excludes takeoff or landing phases).
	F-POST	Fire/smoke (post-impact) Fire/Smoke resulting from impact.
	WSTRW	Windshear or thunderstorm Flight into windshear or thunderstorm.
	ICE	Icing Accumulation of snow, ice, freezing rain, or frost on aircraft surfaces that adversely affects aircraft control or performance.
	EVAC	Evacuation Occurrence where either:(a) person(s) are injured during an evacuation;(b) an unnecessary evacuation was performed;(c) evacuation equipment failed to perform as required; or(d) the evacuation contributed to the severity of the occurrence.
	SEC	Security related Criminal/Security acts which result in accidents or incidents (per the International Civil Aviation Organization (ICAO) Annex 13).
	CABIN	Cabin Safety Events Miscellaneous occurrences in the passenger cabin of transport category aircraft.
	AMAN	Abrupt Manoeuvre he intentional abrupt maneuvering of the aircraft by the flight crew.
	LOLI	Loss of lifting conditions en-route Landing en-route due to loss of lifting conditions.
	UIMC	Unintended flight in IMC Unintended flight in Instrument Meteorological Conditions (IMC).
	GTOW	Glider towing related events Premature release, inadvertent release or non-release during towing, entangling with towing, cable, loss of control, or impact into towing aircraft / winch.
	EXTL	External load related occurrences Occurrences during or as a result of external load or external cargo operations.
	MED	Medical Occurrences involving illness of persons on board the aircraft.
	NAV	Navigation error Occurrences involving the incorrect navigation of aircraft on the ground or in the air.
	UNK	Unknown or undetermined Insufficient information exists to categorize the occurrence.
	OTHR	Other This category includes any occurrence type that is not covered by any other category

Table 2.3: ADREP Occurrence categories

2.2.4 A typology of taxonomies

Table 2.4 summarizes the main characteristics of the different taxonomies organising occurrence data. The typology refers both to the taxonomy used and the specific corpus we have studied. Our only example of the ADREP taxonomy is the DGAC corpus. Thus, the information here does not apply to every collection organized in ECCAIRS.

We distinguish the following characteristics:

- **Structure:** How are the attributes organized. A **flat** structure means that all attributes are on the same level. A **hierarchical** structure means that some of the *values* are organised in a tree-like fashion. *Entity*-based means that both *attributes* and *values* are hierarchically organised. ADREP has a **complex** organisation as entities are interdependent and crosslinked.
- **Inference:** Whether expert reasoning is needed to infer some of the information in the coded data.
- **Independent coding:** If an entity other than the producer of the report is responsible for coding the metadata. ASRS has **some** independent coding, meaning that the producer also fills in some of the attributes directly in the submission form.
- **Cover:** What proportion of the documents are sufficiently coded.
- **Detail:** What proportion of the information about the event is (potentially) representable by the metadata.

Collection	Structure	Inference	Independent coding	Cover	Detail
ASRS	Entity	Yes	Some	Medium	Medium
CADORS	Hierarchical	Yes	Yes	Full	Low
TSB BEA NTSB	No taxonomy	-	-	-	-
Aviation Herald	No taxonomy	-	-	-	-
ASN	Flat	No	No	Full	Very low
ASN Wikibase	Flat	No	No	Full	Very low
DGAC	Complex	Yes	Yes	Low	High

Table 2.4: Typology of taxonomies

2.3 Using occurrence data

In this section we will take a closer look at the way stored occurrence data is used. We can discern three high-level scenarios:

- **Querying the collection:** a person working on some issue is searching for records matching some set of criteria. For example looking for runway incursions in foggy conditions.
- **Producing statistics and KPI**³³: Using the data set to produce statistics about a particular type of event. For example, chart the number of runway incursions per month on a particular airport.
- **Monitoring the system:** Using the data to identify novelty, emerging trends or, more generally anything out of the ordinary that might need attention.

2.3.1 Querying the collection

Querying the collection involves searching for records of occurrences matching a given information need. A user expresses the need via a *query* and the system returns a set of documents matching the query. This is without doubt the most important use made of stored incident and accident reports.

The simplest imaginable query is that for a record, for which the user knows the reference, filing number or other means of unique identification. In such a case the system does not need a lot of complexity, beyond that of simple storage and retrieval capabilities.

More often the information need is more articulate. The following example comes from the ASRS website tutorial on how to use the web-based search functionalities provided by the service :

A flight manager for an air carrier notes that the number and severity of runway incursions at several major airports his air carrier services appear to be down over the past several years. He feels that the runway safety training his and other airline conduct, and the work of FAA's Office of Runway Safety has had a positive impact. Reviewing his airline's training material he decides to update the runway safety training material with more recent ASRS Database examples of runway incursion incidents from the past 4 years.

In this scenario the information required is complex. Isolating relevant occurrences from the database depends on the information available in the taxonomy and how it is indexed. The ASRS tutorial continues and points out the three fields the user needs to query in order to obtain a subset of records.

³³Key Performance Indicator

Date of Incident = January 2004 – December 2008
Federal Aviation Regs = Part 121
Carrier Location = BOS.Airport, LAX.Airport, ORD.Airport,
DFW.Airport, ATL.Airport, IAD.Airport
Event Type = Ground Incursion, Runway

A further refinement of the query is needed in order to narrow down the result by searching in the narrative fields, using the (limited) full-text indexing capabilities provided by the ASRS search engine.

the flight manager notes that there is a wide spectrum of causal and contributory issues in this data set. He really wants to focus on incidents where confusion or misunderstanding played a role, so he modifies his search strategy.
Step 2: Do not change any of the values for Date of Incident, FAR Part, Location, or Event Type. Add the following text search terms:
Text = Confus% OR Misunder%

Note: The “%” symbol will find all words where the text begins with what was entered, i.e., “Confus%” will find “Confusion,” “Confused,” etc. The “OR” operator will surface records that reference any of these terms. Make sure both “Narrative” and “Synopsis” are checked.

This example shows how metadata and full-text search are combined in order to answer the information need of a user. Given the complexity of the query, GUI³⁴ solutions need to be adapted to the query. Figure 2.26 shows the query-builder on the ASRS website. A query is formulated by first selecting the entities one is interested in. Then, for each entity, a separate pop-up window appears in which the user either chooses the value from a list or types in a string.

In the next chapter (§3.1) we will discuss solutions to exactly this type of issues from the field of information retrieval.


2.3.2 KPIs and statistics

Another common use of collections of incident reports is producing statistical information and key performance indicators. This use relies on aggregating information for multiple records in order to gain insight into the global state of affairs at any given time.

Regulators, such as the French DGAC annually publish a report with a panorama of air safety throughout a given year. In their 2013 report (DGAC, 2013), the DGAC reassure the public that air travel is getting safer every year.


³⁴Graphical User Interface

How To Search:


Step 1: Click  to add search items. Note: Make sure your Pop-up Blocker is off.


Step 2: In "Current Search Items" section, select "Click Here" in a statement and choose items from lookup window.


Date & Report Number

 **Report Number** (ACN) was [\[number\]](#)


Environment


 **Flight Conditions** were [\[conditions\]](#)


 **Lighting** was [\[conditions\]](#)


 **Weather** was [\[element\]](#)

Aircraft


 **Flight Plan** was [\[type\]](#)

 **Flight Phase** was [\[phase\]](#)


 **Make/Model** was [\[aircraft type\]](#)


 **Mission** was [\[operation\]](#)

Place


 **State** was [\[abbreviation\]](#)


Person


 **Reporter Organization** was [\[type\]](#)


 **Reporter Function** was [\[position\]](#)


Event Assessment


 **Event Type** was [\[anomaly\]](#)

 **Detector** was [\[equipment/human\]](#)


 **Primary Problem** was [\[most prominent factor\]](#)

 **Contributing Factors** were [\[problem areas\]](#)




 **Human Factors** (since 6/09) were [\[factor\]](#)

 **Result** was [\[consequence\]](#)

Text: Narrative / Synopsis

 **Text** contains [\[words\]](#)

Current Search Items:

-  **Date of Incident** was between [January-2004](#) and [December-2008](#)
-  and **Federal Aviation Regs** (FAR) Part was [Part 121](#)
-  and **Location** was [BOS.AIRPORT or LAX.AIRPORT](#)

Back
Run Search

Figure 2.26: ASRS search GUI

Figure 2.27 (DGAC, 2013, p. 12) shows that there are fewer fatal accidents, both per million departures (grey line) and per billion kilometres travelled (green line) as well as fewer individual fatalities per billion kilometres (black line).

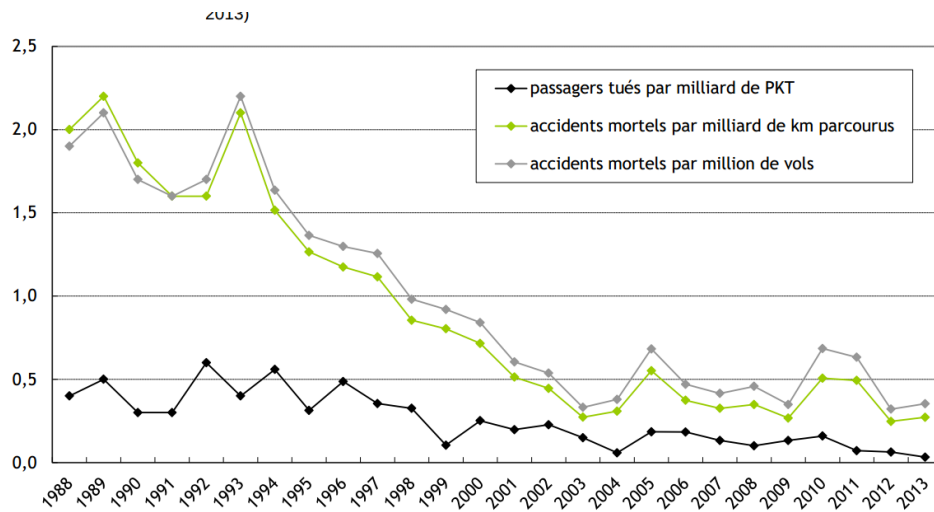


Figure 2.27: Fatal air accidents in France since 1987

Detailed breakdowns for different cross sections of the industry are also provided. Figure 2.28 (DGAC, 2013, p. 30) gives a typology of accidents in 2013 of general aviation aircraft registered in France. The criterion used is the *occurrence category* of the ADREP taxonomy (§2.2.3.5). One can see, for example, that most fatal accidents result from loss of control in flight³⁵, or that abnormal runway contact accidents³⁶, while being of the most frequent type did not cause loss of life.

³⁵perte de contrôle - en vol

³⁶contact anormal avec la piste ou le sol

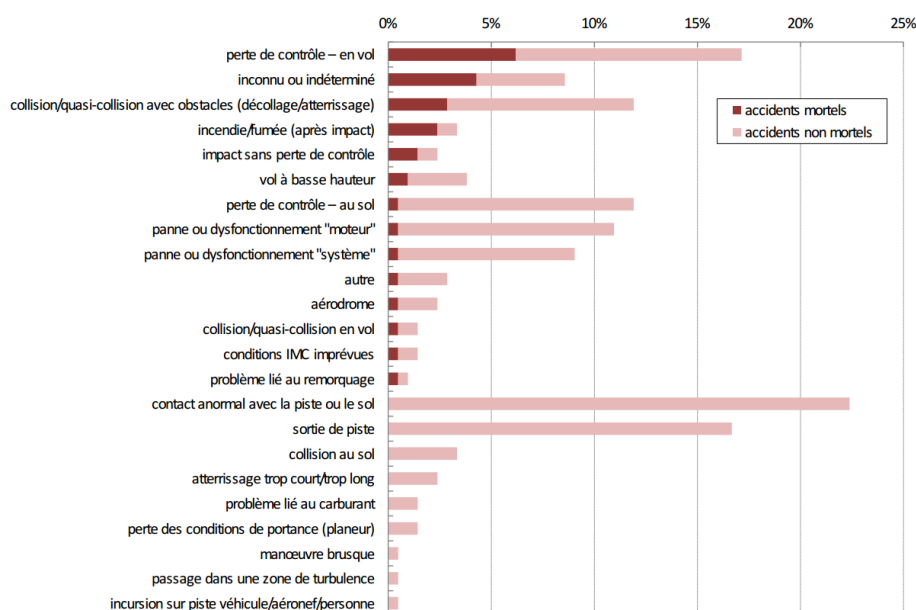


Figure 2.28: General aviation accidents in France in 2013

The required precision of the underlying data needed to produce such results varies between the two. While only tracking the number of accidents and the number of fatalities is sufficient for the first example, a far more detailed description is necessary for the second. In order to produce coherent statistics, one needs to accurately classify individual occurrences with a much finer grain size. Maintaining coherent fine-grain descriptions, as we will see in the next section can become problematic when huge amounts of incident data is tracked.

There is no limit on the potential finesse of description needed when performing aggregated analysis for a given type of risk. The risk matrix approach, for example is a well established decision-making protocol, where the frequency of a given type of incident is compared with its severity to produce a unified metric of risk. The rationale behind this approach is that relatively benign incidents can be allowed to happen frequently with less risk, where more consequential ones should not occur often. Risk matrix analysis allows directing corrective actions in an informed manner.

Allan (2006), for example, performed a risk matrix type analysis of bird-strikes and communicated thresholds of bird activity *per bird species*, over which airports need to take action in order to both minimise risk and optimise the resources involved with bird control. The success of this approach hinges

on the availability of large quantities of incident data regarding birdstrikes, where the specific species of bird is available as an aggregation criterion. The same holds for any type of risk matrix analysis. The absolute minimum requirement is having reliable data in the first place.

When we factor in the inherent noisiness of the data, we can easily see that lack of coherence can skew the results. Allan's 2006 study would not have been possible if data about species of birds colliding with the aircraft was not tracked, coded and stored in the first place.

There are however ways to extract the needed information from data, in which the information is contained, but not in a readily machine digestible form. Bird-strikes are a good example. In our experience with incident reports, we often see the precise species mentioned in the free text narrative. This information is easy to extract and then process for purposes such as the aforementioned study and in Section 3.2 we show how supervised learning techniques can produce reliable classifications for bird-related incidents.

In this sense we can almost see the problem of producing KPIs and statistics as a more complex version of querying the collection, where one looks not only of examples of a given type of event but of *all* the concerned occurrences and where noise ought to be kept at minimum.

2.3.3 Intelligence and monitoring

This brings us to the third use of incident and accident data. As Johnson (2003, p.735) explains:

Identifying trends. Databases can be placed on-line so that investigators and safety managers can find out whether or not a particular incident forms part of a more complex pattern of failure. This does not simply rely upon identifying similar causes of adverse occurrences and near misses. Patterns may also be seen in the mitigating factors that prevent an incident developing into a more serious failure. This is important if, for example, safety managers and regulators were to take action to strengthen the defences against future accidents.

This type of use essentially bridges the previous two. Starting from a specific incident the expert will query the collection in order to find others that resemble it. He will then perform some measure of aggregation, whether purely statistical (as in counting) or informational, where he will interpret and summarise his findings. It was already question of such a use in Section 1.3, where experts trace a series of related events to a common source.

This type of use relies the most on the human expertise involved, as interpretation by an expert is an integral part of the process. One should however seek to provide them with the best tools suited for the job. This reasoning

motivated us to research and develop the *timePlot* system for identifying similar reports, which we present in Chapter 5. Identifying the particular ways in which individual occurrences relate one to another is a viable solution to aid experts to discover trends.

Intelligence and monitoring, aided by heavily automated components will surely be part of the systems of the future. We can not but recall here one of the original (and overambitious, as is often the case) goals of this thesis - building automatic monitoring systems. Factoring in, on the one hand the dynamic nature of activities, one can not only look for trends, but to monitor in their development over time, identifying relevant occurrences as they happen. From here it is a small step to imagine anomaly detection components that automatically identify trends as they start developing, based on disturbance of the temporal distribution of events in the information flow. In our case, we quickly came to realise that such systems will only be possible once we find ways to deal with the noisiness of the data in order to provide inputs of sufficient quality. Thus our focus shifted to the preconditions. We nevertheless continue to consider automatic monitoring systems as a long term objective and direction of future research.

Fuelling this use of incident data is a general state of affairs worth mentioning. We have dubbed it the ‘*collective hindsight bias*. Hindsight bias is “is the inclination, after an event has occurred, to see the event as having been predictable” (Roese and Vohs, 2012). In the case of high profile accidents there is a comparable effect on a social level. The mix of emotions focus attention on the one hand to the experts responsible for preventing the disaster and on the other prompt “all available hands” go sifting through the record to identify a trend or a signal that *should have* alerted someone. With enough manpower someone always finds a signal, a trend that lead to the catastrophe. If it holds at least a little bit of credibility this is picked up by mainstream media. In the but rarest of cases is the trend a genuine type 2 error, but nevertheless the institutions have no choice but to spend valuable resources explaining or debunking the hypothesis. The fear of such a situation puts extreme pressure on safety experts not to miss a pattern in the data.

2.4 Issues when dealing with large collections of occurrence data

The uses we described in the previous section all depend on available and high-quality data. Yet as we will see this is not the case in reality. We have identified a number of issues that hinder the proper execution of those tasks. In a nutshell, when all the information is present in the form of natural language, it is nearly impossible to exploit in an adequate manner. For this reason the industry makes heavy use of taxonomies but those are far from perfect and suffer from their unique set of problems.

2.4.1 Issues with natural language reports

As Johnson (2003, § 16.2.4) points out:

It can be difficult to detect patterns of failure amongst the natural language accounts of adverse events that are produced by many reporting systems. The volume of prose produced in national and international systems can make it difficult for any individual to keep track of common causes or consequences across many incidents.

When incident and accident reports start to pile up, they become notoriously difficult to exploit without the adequate tools. Full-text search engines are becoming ubiquitous and many off the shelf solutions exist, yet as the example in the previous section (§2.3.1) illustrates even serious institutions such as NASA, maintaining the ASRS database struggle to provide full-text search capabilities that take into account the inherent variability of natural language even at a basic level.

Official accident reports, most often published in *pdf* format are sometimes impossible to query without processing and some publishers only provide the most basic keyword search capabilities.

The NTSB website's integrated search for example only allows to search for contiguous strings of words in the text. It does not account for even basic variation such as plurals, provides no term highlighting and searches via a sequential scan of their whole database at each query, considerably slowing down the process.

Obviously, producing reliable statistics from such reports is also impossible without first manually classifying them into whatever aggregation criterion one is looking for. If no metadata is present, in order to produce the example from the previous section (fig. 2.28), an expert would need to comb through all the reports and classify them within an occurrence category schema.

Looking for patterns and monitoring the system also would require reading all incoming reports and manually tracking them.

Multilingual databases also pose their unique set of problems. While English is the *lingua franca* of aviation, other languages are sometimes also used. This is an issue when maintaining large databases of incident reports and seeking to access their content. Large national airline companies often collect data in both English and the local language in their internal reporting program as many of the pilots are not native or bilingual. Accessing the data is problematic and we witness different in house solutions developed for convenience, such as translating the titles of the reports so that they are all in the same language (fig. 2.14).

For the above mentioned reasons, in practically all cases some classification is performed. Relying on taxonomies though comes with its own set of problems.

2.4.2 Issues with coded data and taxonomies

As we saw in the previous chapter, accurately describing an occurrence is a complex and expertise-intensive task. Even the simplest of incident reporting system may have tens of fields for factual data and at least one set of high-level categories. More complex ones such as ICAO's ADREP taxonomy used in ECCAIRS have thousands of fields describing every aspect of the occurrence. Such systems are designed with scale in mind. The initial ambition of these systems is that, by using taxonomies, they will constrain every possible occurrence within a predefined set of possible values with minimum loss of information. "An uncoded occurrence is a lost occurrence. It is unusable. It is just dead weight in the database" once told us an expert working with ECCAIRS data sets, meaning that the quality of coding is paramount to the success of data-based risk assessments. In reality however, this is rarely the case for various reasons.

2.4.2.1 Complex codification schemes

Codifying each incident is a time consuming task which requires domain knowledge and expertise to perform. A considerable effort is needed to perform it consistently for every occurrence and in some cases codifying each occurrence is simply abandoned. The DGAC's database contains 404289 occurrence reports from 2004 until September 2014. From these, roughly a third (136861) are coded with the *occurrence category* attribute, the most simple high-level classification of the ADREP categorisation schema, consisting of only 36 values on a single level.

Such large and omnipotent taxonomies are so complex that coders are required to undergo training in order to use it. Specifications of when to use and when not to use a given class can be vague and coders might not fully understand them or interpret them differently. A given occurrence, it follows will not be coded in a identical manner depending on who coded it. Johnson (2003, pp. 767-768) mentions that a 75% rate of accuracy in trainee coders is acceptable for graduation.

When working with aggregated ECCAIRS data we have even seen "dialects" appear. Groups of coders (usually in the same company or department) agree between them on how to interpret the taxonomy, thus producing coherent data which however differs from the data coming in from other departments or companies.

2.4.2.2 Dynamic systems and static taxonomies

While the above-mentioned practical issues may be overcome in theory (by say hiring coders in mass), there are however more fundamental ones when relying solely on coded data. The system is constantly changing and safety is a lot about responding to such change and dealing with novel and unseen

combinations of factors before they start resonating and create an unsafe state (Hollnagel, 2004). A taxonomy, on the other hand is by definition a static structure that is designed to accommodate occurrences at a given time. It follows naturally that novelty will be difficult to account for in a taxonomy that was designed with occurrences predating the novelty, perceiving novelty is of critical importance when assessing risk.

Given that, depending on the scale of the system, the update cycle of the taxonomy might be quite long (as is the case with ECCAIRS) a sub optimal solution will be required to code those occurrences where the novel elements are present before the novelty is introduced in the taxonomy.

2.4.2.3 Changing models and taxonomies

The previously cited issue has also an inverse implication. When change is introduced in the taxonomy it is sudden while the novelty it accounts for is gradual. It follows that the occurrences collected before the updated taxonomy will inevitably be coded using whatever sub optimal solution was applied while those collected after the new version is introduced are codified with the new classes. This creates a completely artefactual shift in the (already noisy) data.

Novelty is by far not the most extreme case. In the ECCAIRS world changes are introduced gradually and, while there may be some lag, the data still is fairly consistent. In other cases however, in particular with less formalised incident reporting solutions of a smaller scale, such as localised internal reporting systems, paradigm shifts in the very way of accounting for risk imply whole-scale overhauls of the models used, a completely novel taxonomy and a change in the supporting software solutions.

Such is the case at this moment with a number of airlines that build internal safety management systems. It follows that at the time the system is introduced, all occurrences collected before that date are completely incompatible with the new way incident reporting is implemented.

As narrative parts are unaffected by changes in models and taxonomies, they provide a valuable resource for retrofitting occurrences to a newly introduced model using automatic text categorisation (§3.2).

2.4.2.4 Bottleneck effects

Bottleneck effects are specific issue whenever aggregating data is concerned at the level of (inter)national regulators or at the level of companies working with various external data sources.

Data originating from structures with varying takes on incident description, using different accident models and taxonomies and employing different software solutions to manage them, generates a stream with many levels of fragmentation. Usually the receiving entity employs their own occurrence management solution and taxonomy and must accommodate the stream of

occurrence data into their system. Rectification and transformation layers, custom ETL³⁷ solutions and (where applicable) conventions on the exchange formats and content provide part of the solution but at the end the data is often impoverished by the transfer and exchange process. Moreover, contrary to the change-of-taxonomy scenario, in this type of situations the data is sometimes coherently codified and thoroughly analysed at the source. Valuable expertise is thus wasted due to format incompatibilities.

Given that collation of reports implies an increase in volume, the need for adequate data management and analysis solutions is proportional. Narrative parts of documents are relatively³⁸ “immune” to bottleneck effects and are transmitted. In the most extreme of cases they vehicle the major part of the information contained within the reports.

2.5 Summary of the issues and NLP as a solution

In the previous chapter we saw how feedback information gathered through incident reporting plays a vital role in ensuring the safety of any given system. In this chapter we saw in particular how this data flows and is used in the civil aviation community as well as how the data is managed and stored. In order to summarise and better present the challenge we are addressing, let us first take a step back and imagine what a perfect world would look like. In our utopia:

- All incidents will be reported worldwide on time with all the information about the occurrence will be reflected in the report.
- Anyone who is interested will have access to this body of information.
- The data will be stored in a single common format.
- The data will be organised (via a taxonomy) and indexed in such a way so that a query could be formulated to (fully or partially) describe any accident or incident scenario, complete with any level of detail regarding the occurrence’s context.
- Any grouping on any criterion could be performed in order to produce aggregations and perform quantitative analysis such as statistics trends.

Basically the utopia boils down to two points: Collecting all relevant data (which is far from the scope of this work) and providing powerful means for accessing the information it contains. In a way the second part is already being addressed by the ADREP taxonomy, which (while far from perfect) in

³⁷Extract Transform Load

³⁸There are cases where narratives are split into different sub fields, each with a different discursive function that need to be collated into a single one. When gathering ASRS reports (2.1.4) for example the host solution might not have the structure needed to accommodate multiple narratives and a separate synopsis.

itself remains a considerable advance with respect to most other industries. In theory every accident could be coded in ECCAIRS using ADREP and then be available to be part of the answer to whatever safety related question somebody asks. In practice this is not the case. The example we saw in section 2.2.3.5 on the perfectly coded Concorde crash, came from a PowerPoint presentation of the tool. In reality we have never seen such a well coded occurrence in all the years of working with such data. The system simply does not scale without investing unbelievable resources for consistent manual coding.

What ADREP really attempts is to normalise this information in a schema sufficiently abstract to be generalised over a large collection, but yet sufficiently precise to capture most of the intricacies of speaking about flying airplanes, hence its complexity. However one tends to forget that most of the information available in an incident or accident report is also contained in the text. Just compare the natural language and ADREP versions of the Concorde crash (figs. 2.24 and 2.25). Almost all the information from the ADREP version is in the probable cause statement.

So what is its status of natural language narratives in the present?

For one, efforts at developing and maintaining taxonomies and adopting coding-based solutions has had the effect to occult solutions aimed at exploiting the narrative parts. The predominant rhetoric seems to be that (especially for incidents) natural language accounts are for human consumption only. They are read at the time of collection and screening, but once the process is over and the immediate actions taken, the report is archived in a database. The text only becomes useful if a human reads it once more. But ironically, natural language accounts are the most resilient bit of data in an accident report as it flows through the system. They are immune to bottleneck effects, to format changes and to taxonomy incompatibilities. and they are ubiquitous - 99% of accident and incident reports have narratives.

So the question we ask is: **how, by considering report narratives as an input material, can we provide safety experts with better access to the information that incident and accident reports vehicle?**

On one hand, if we follow the current trend in the industry, we might consider that taxonomy based approaches are the only way to go and then search for methods that help to more efficiently code and maintain the data using narrative parts as input material for automatic classifiers. We discuss such work in section 3.2.

On the other hand, we could consider that narratives are all that is needed. To put it in other words, to declare that taxonomy based approaches are a failure and start researching how to build the most powerful full-text search engine to replace them.

Balancing between these two extremes we mostly explored the relationship between text and metadata in an empirical manner. In the next part we will show that, by considering natural language as raw input material to various

processes, many of the issues presented above can be addressed, improving both the overall coherency of the data as well as the specific modes of accessing it required by the industry. More specifically:

- The first and foremost need we identified with safety experts is to **identify patterns** in the data (§1.3, §2.3.3). Our first take on the subject was to build a system detecting similar occurrences in large incident databases. In Chapter 5 we describe a system used to detect similar occurrences in large incident databases. Today the system is used in a large airline (§2.1.5) and at the DGAC. A demonstration version, soon to become a commercial product is also available for ASRS (§2.1.4 and The Aviation Herald (§2.1.6). The system is used also as a full text search engine, providing a much simpler and more intuitive solution than ECCAIRS and the search engines provided by the web services of the data providers. Furthermore, willing to explore the taxonomy/text informational redundancy, we experimented using the DGAC's database, looking to neutralise aspects of variation already captured by ADREP *Occurrence Categories* in order to find reports that are similar for reasons absent in the original coding. This work is presented in Section 6.2 and is a first step towards devising methods that search for secondary patterns, hidden by the primary aspects of variation.
- The need to **identify particular occurrences** (§2.3.1) is broader than just typing in a query. While a particular incident scenario is easily constructed in an expert's mind, it can be impossible to express as a database or search engine query. For this reason, in Section 6.5 we present experiments with active learning, a technique using supervised classification as a constant input from the users. This system uses a query as a starting point and then allows the user to further refine the model by validating pertinent or invalidating impertinent occurrence reports. We run a simulation using both the DGAC's data and the ASRS database and show that such an approach can be used to model a particular facet of an incident for which for one reason or another is absent from the coded data.
- The need to **automatically classify reports** in a given taxonomy is straightforward (§2.4.2). In Section 3.2 we present a system using supervised classification to automatically produce ADREP occurrence categories for the DGAC's database, which collects between 4000 and 5000 reports every month from all the service providers in France, leading to problems such as bottleneck effects as well as maintenance issues when the taxonomy is updated.
- Multilingual databases are an issue for large service providers and for entities that collect incident information from different sources (§2.4.1).

In Section 6.3 we present a system inspired by Cross Lingual Explicit Semantic Analysis (Sorg and Cimiano, 2012) in order to detect similar reports written in different languages (English and French) we use the data provided by the Canadian accident investigation authority (§2.1.2.2) and evaluate it on the CADORS database (§2.1.3).

- Finally we address the need to **construct a taxonomy from scratch**. While it is not expressed in civil aviation, it can be an issue in many other industries. A collection of incident reports may only contain narratives. So, in Section 6.4 we present work that evaluates the application of *Topic Modelling* on the ASRS database. We compare the *topics* produced by this method to the metadata. We discover first, that *Topic Modelling* can identify facets of incidents that are not present in the taxonomy and second that there is a considerable overlap between the metadata and the extracted *topics*. This proves that the method is a valid starting point for constructing taxonomies in a data-driven manner.

2.6 Chapter conclusion

In this chapter we saw the different types of occurrence data, how it is produced, stored and used for ensuring safety in civil aviation. We saw that access to information in large collection is paramount to the process and identified those areas where the current *taxonomy*-based paradigm fails to meet expectations in real-life usage scenarios. While natural language is omnipresent in occurrence data, the existing tools do not allow easy searching within the narrative parts. In order to access the information, one has to rely on taxonomies, but those are not always adapted, being either too broad to describe the event or too complex to sufficiently cover the collection.

In the next chapter we will present the domains of NLP that address the problems of searching in large document collections and automatically classifying documents based on their textual content and discuss how these domains potentially improve access to information in collections of incident and accident reports.

Chapter Three

NLP: domains of application

This chapter presents the domains of Information Retrieval and Text Categorisation. Each section is organised by first presenting the domain and the key concepts, before discussing the specific implication of their application to occurrence data.

We saw in the previous chapters that safety today is dependant on accessing information which is contained in a variety of document types. These documents are stored in collections with different modes of organisation. They can be anything from flat lists of *pdf* files to databases organised within highly complex taxonomies, structuring large lists of metadata attributes.

The information is used for different purposes, from exploratory browsing and looking for patterns to precise querying for quantitative assessments. The issues we listed in Section 2.4 boil down to a single question.

How do we provide efficient means for finding information in these collections?

There are two ways to look at the problem. On one hand, a lot of the information we are looking for is in the natural language parts of occurrence reports, we then turn to field of Information Retrieval, the discipline involved with building search engines. On the other hand, metadata is the *de facto* standard way of accessing and exploiting these documents. The quality of the coding and the availability of coherent metadata are however far from satisfactory. For this reason we turn to the domain of Text Categorisation for exploiting the redundancies between text and metadata and automatically enhance the quality of the coding.

3.1 Information retrieval

Today search engines are ubiquitous and are taken for granted. From Google to custom industrial solutions used to query highly specific data sets, these systems provide a solution to our need to be able to find information in a collection of documents. Information Retrieval is the academic field concerned with building search engines, or as Manning et al. (2008) put it:

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

3.1.1 Problem definition

The standard way of defining the problem is remarkably simple. The user has an *information need*. He formulates the information need via a *query*. The system analyses the query, *matches* it with the documents contained in its index and *returns* documents containing the information that satisfies the user's need. This way to look at the problem is pretty much stable and is a reference in the field (Robertson, 1977; Manning et al., 2008).

Figure 3.1 (diagram by (Chevalier, 2011)), presents the U-shaped process of information retrieval as described by Salton and McGill (1986).

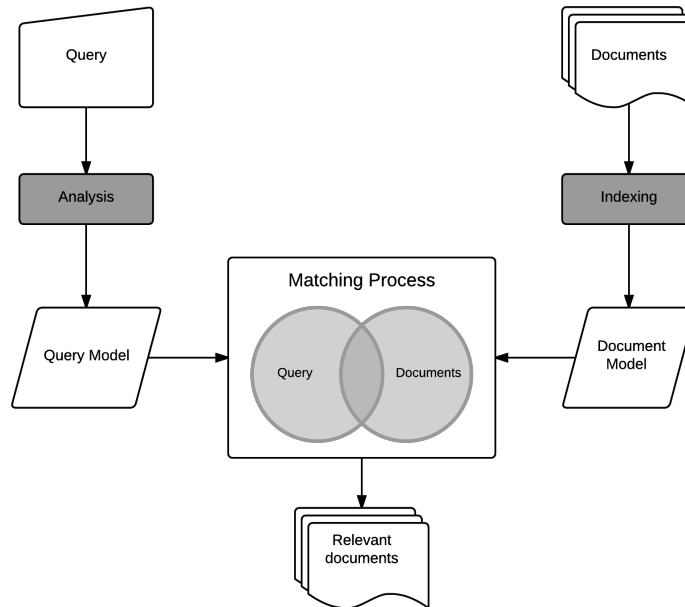


Figure 3.1: The IR process

Four key phases are part of the Information retrieval process:

- **Expression of the information need:** The process by which a user transforms a mental need into a *query* that he submits to the system.
- **Document processing:** When documents are added to the system they must be analysed and stored in such a way as to allow their content to be available to the users and answer their information need. This phase is also called *indexing*.
- **The matching process:** The identification of relevant documents that answer a certain information need. The *query* is compared to the *index*.
- **Displaying of the results:** The system returns the relevant documents and presents them to the user.

3.1.1.1 Information need and query formulation

Marchionini and White (2007) define the process of query formulation as a two stage process. The user first **formulates** his problem and then **expresses** it to the information retrieval system. “[The] formulation activity follows acceptance and involves the information seeker conceptualising the bounds of the information need, imagining the nature and form of information that will

meet the need, and identifying possible sources of information pertinent to the need.” Depending on the search system (library, web search, database) the user will employ a different strategy to express his information need. The query, in this sense can be, for example a face-to-face conversation with a librarian or an expert, a full-text query in a search engine or an SQL query in the database. The expression step is thus highly dependant on the particular system’s *interface*.

3.1.1.2 Models of IR for document processing

In the IR process, the phases of *Query analysis* and *Indexing* are consistent with the transformation of documents and queries into a common format in order for them to be matched at the matching stage. There are currently two frameworks for performing this operation: the *Boolean model*, the *vector space model*:

- **The Boolean model:** This is the earliest and most simple framework for IR. Manning et al. (2008) introduce by the problem with a mock example:

A fat book which many people own is Shakespeare’s Collected Works. Suppose you wanted to determine which plays by Shakespeare contain the words Brutus AND Caesar and NOT Calpurnia.

This is an example of a Boolean query, as the information need is formulated as a Boolean expression, where the result is a subset of documents from the collection satisfying all the constraints. Boolean search is still preferred in some cases as it allows for very precise queries and results.

- **The vector space model:** Unlike Boolean retrieval, which is very precise, *vector space* (Salton et al., 1975) modelling allows to quantitatively represent the relationship between documents and queries and the attribution of a score of relevance to the returned results. Thus the results can be ranked according to how well they resemble the query (and answer the information need) Vector space modelling is the predominant paradigm in IR today and will be discussed in detail in the following chapter (§4.2).

3.1.1.3 Displaying the results

The end results of the information retrieval process is to display relevant documents to the user. The most common display mode for retrieved documents is the *ranked list* with *snippets*. A snippet is an excerpt from the document showing the parts that correspond to the query. Usually this is done via some form of highlighting, like bold face formatting of the query terms. Depending

on the specific search task, many other display modes are possible and have been applied (Kules et al., 2008).

3.1.1.4 IR performance

A perfect IR system will return *all* the relevant documents to a given query and *only* those documents. However this is rarely the case. There is always some irrelevant documents in the results (noise) as well as some relevant documents are missing from the results (silence). Combining the notions of relevance and presence in the results, the documents during a given search session can be split into four categories:

- **True Positives:** Documents that **are** relevant and **are** returned by the system. (expected)
- **False Positives:** Documents that **are not** relevant and **are** returned by the system. (noise)
- **False Negatives:** Documents that **are** relevant and **are not** returned by the system. (silence)
- **True Negatives:** Documents that **are not** relevant and **are not** returned by the system. (expected)

A perfect system thus produces no false negatives and no false positives.

In order to measure these parameters and evaluate IR systems the metrics of precision and recall are used. Manning and Schütze (1999) define the two as:

- Precision (P) is the fraction of retrieved documents that are relevant.

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{overall relevant items})} = P(\text{relevant}|\text{retrieved})$$

- Recall (R) is the fraction of relevant documents that are retrieved.

$$\text{Recall} = \frac{\#(\text{relevant item retrieved})}{\#(\text{overall retrieved items})} = P(\text{retrieved}|\text{relevant})$$

The higher both values are the better the system performs.

We will now use an example and present some of the issues that arise when searching information in natural language documents.

3.1.1.5 An example of full text search problem

The predominant mode of interaction with a search engine are *free text queries*. The information need is expressed with a string of words without the user needing to concern himself with the exact underlying Boolean logic. From

a user's point of view the process looks simple. Just type in what they are looking for. However, the relationship between the query and the document is not always that simple. Let us walk through a real example of a search scenario and look at the basic language-related issues.

Imagine the following information need:

From a collection of incident reports published by the Aviation Herald (§2.1.6), an expert is interested in cases where the aircraft slid off a wet runway.

The term denoting this type of incident is a “runway excursion”. If it were coded in the ADREP taxonomy, it would have had “RE:Runway Excursion” as its *Occurrence Category* attribute (§2.2.3.5). Knowing this, a user will formulate a query like “wet runway excursion”. If we look at this query as a Boolean expression, the subset of documents returned can be defined as all documents containing the words “runway” AND “excursion” AND “wet”. In this set some documents will satisfy the user's need for information.

Accident: Iran Aseman F100 at Tabriz on Aug 26th 2010, runway excursion into a ditch

[...] The airline confirmed the airplane suffered a runway excursion reporting some damage to the nose section of the aircraft. No injuries occurred. [...] Reportedly the landing was performed in wet and windy conditions on a wet runway with one thrust reverser deactivated. [...]

Figure 3.2: AH report 43028227

Accident: Trans States E145 at Ottawa on Jun 16th 2010, runway overrun

[...] - The smooth landing on a wet runway led to viscous hydroplaning, which resulted in poor braking action and reduced aircraft deceleration, contributing to the runway overrun. [...]

Figure 3.3: AH report 42d059bc

Figures 3.2 and 3.3 are excerpts from documents that describe wet runway excursions. Both are relevant (*true positives*) and satisfy the Boolean constraints.

There are however reports that do not satisfy the constraint and yet satisfy the user's information need. An example is the report in figure 3.4, which is a *false negative* and part of the *silence*.

Incident: Delta Airlines DC95 at Savannah on Sep 27th 2010, overran runway

A Delta Airlines Douglas DC-9-50, [...], was on approach to Savannah in thunderstorms, heavy rain and difficult wind conditions. The airplane landed [...] but overran the end of the runway coming to a stop within the paved surface of the overrun area.

Figure 3.4: AH report 4317a108

This is a classical issue of **lexical variation** - different words sharing similar meanings. An *overrun* is a specific type of (*runway*) *excursion*. From a linguistic point of view we will consider that these words are pseudo-synonyms, but the issue extends beyond synonyms. The example in Figure 3.4 does not contain the word “wet” but does contain “heavy rain”. The two are clearly related and from a system’s point of view, such a relationship must be accounted for in order to improve recall.

Another issue is that of **compositionality**. Meaning is lost when we take words out of context. The example in Figure 3.5 illustrates a report that contains both the words “runway” and “excursion” but is clearly irrelevant¹ to the query as it does not concern a “runway excursion”. This document is a *false positive*, an example of *noise*

Report: Flybe DH8D near Southampton on Mar 3rd 2009, stick shaker activation and temporary loss of control during turn onto base

[...] While turning onto the base leg for runway 20 the stick shaker briefly activated due to turbulence, the autopilot disconnected, the airplane pitched up to 12.5 degrees nose up and rolled to a 43.5 degrees left bank before the crew was able to regain control. [...] When the autopilot disconnected she put her hands back onto the control wheel and felt the stick shaker for a moment. She attributed the following attitude excursions to turbulence.

Figure 3.5: AH report 42cbc93c

Finally the example (*false positive*) in Figure 3.6 concerns a report that makes reference to another report. The search terms are situated in the reference and not in the main text. It illustrates how **discourse structure** can have an impact on the relevance of the results.

¹For this example we simplify the information need and omit “wet” from the criteria.

Incident: SATA A320 at Lajes on Mar 4th 2011, ran over obstacle on runway

The runway was wet with runway markings hardly recognizeable, re-surfacing work was in progress. [...] [D]uring roll [the aircraft] ran over a sand bag holding a cable for the temporary runway edge lights causing damage to a tyre, that did not deflate however. [...] Following another similiar incident new NOTAMs were released on Mar 11th clarifying the work in progress and runway modifications, see also Incident: Travel Service B738 at Lajes on Mar 10th 2011, runway excursion on landing.

Figure 3.6: AH report 439534d5

3.1.2 Linguistic issues in IR

The main drawback most IR techniques suffer from, as Arampatzis et al. (2000) put it, is “that they still make the assumption, that if a query and a document have a (key)word in common, then the document is about the query”. Language is not merely a bag of words, it has structure, texture and regularities. “[Language] is a mean to communicate about concepts, entities and relation, which may be expressed in many forms”. Linguists understand many of the intricacies of language and how it varies on different levels. Accounting for variation improves the performance of IR systems both in precision and in recall. Variation is omnipresent in language, and to a large extent it can be described and measures taken to normalise the inputs to an IR system (documents and queries) so they exhibit *less* variation at the moment of *matching*, than they exhibit in their raw unprocessed forms.

Let us look at the main types of linguistic variation, their implications to IR and some potential solutions.

3.1.2.1 Morphological variation

Morphology describes the internal structure of words. It is usually broken in two parts, *inflectional* and *derviational* morphology. The first describes changes undergone by words as a result of syntax, while they remain the same part of speech and keep practically the same meaning. A noun will have a plural form (excursion, excursions), a verb will have *inflections* according to person, tense and mode (land, lands, landed).

Derivations are forms that are derived from a certain word usually to form a different part of speech. Nominalisations (wet/wetness and land/landing), for example are created by adding bits to the ending of a word (suffixation). The derived forms usually keep a common aspect of the meaning of the form they are derived from although in some cases it is rather implicit (mark, markings). While English is a morphologically simple language, others such

as Finnish, German and Turkish exhibit a very large number of variants for every given base form.

Not taking into account morphological variation in an IR system hurts *recall* as documents containing variants of the query terms are not retrieved.

Morphological variation can be taken into account in two² ways in IR systems, one linguistic and one non-linguistic:

- One can might employ a *lemmatiser* and derive a base form for all the variants which will then be used to match them. Plural nouns will be folded to the singular, gender to the masculine and verbs to their infinitive form.
- Or one can use a non linguistic method, known as *stemming*, where words are stripped of their suffixes and folded down to a common form, their *stem*.

Both methods have their pros and cons. Stemming is error-prone but robust. A *stemmer* can be applied to any surface form. Because it relies only on surface forms however, it may incorrectly stem some words. “Organisation” for example will become “organ”, or in French “laser” will become “las”³ *Lemmatisation* on the other hand is costly, as it needs a resource that lists all the variants and their common root. Moreover, in order to be correctly performed the *lemmatiser* needs to know the part of speech of every term. For this reason, the *lemmatisation* is associated with *POS-tagging*, the process that derives parts of speech. For occurrence data, with all the domain vocabulary, one practically has to build such a resource from scratch. Costly and difficult as it is, *POS-tagging* and *lemmatisation* are however an essential step if one wants to perform more complex processing such as syntactic parsing.

3.1.2.2 Lexical variation

Lexical variation implies a whole range of phenomena where words vary in form due to reasons unrelated to morphology and, like morphological variation, hurts recall in an IR system. They can be:

²Actually there are three. One might list and add all the variants of all the terms to the query. This is known as *query expansion* but it is terribly inefficient both to perform at every query and to maintain an index with unnormalised terms.

³This particular example plagued us in the *timePlot* system. The *stemmer* treats the string “laser” as the infinitive of a French verb from the first group. Given that laser pointers are a current subject of interest (5.3.3) and LAS is the airport code of Las Vegas, the system returned a lot of false positives due to this stemming error, for which we had to account by introducing lists of stemming exceptions.

- Surface variation: Words written in capitals, varying use of diacritics, varying use of intratoken punctuation marks⁴ (hyphens, slashes etc).
- Spelling errors.
- International spelling variants, such as the -ize/-ise variation of suffixes between American and UK English.
- Acronyms and abbreviations, such as the ones in the ASRS (TKOFF / take-off) (also see Figure 4.1).
- Real synonyms. (accurate / precise).
- Pseudo-synonyms. (excursion / overrun)
- Translations, such as the use of English terms in French texts.

ASRS's IR system's online documentation illustrates the issue with the following instructions:

We strongly suggest that for better text search results you use as many variations of a word as possible including its abbreviation. For example, if a user is looking for reports that reference the word "takeoff" in a reports' body of text, the terms/words "tkof," "take off," and "take-off," should also be included in the search strategy in order to obtain the best possible results.

There are two ways to tackle the problem of lexical variation, a symbolic and a statistical approach:

- Symbolic approaches mean that the information about the equivalences is encoded in an external lexical resources. *Wordnet* (Miller, 1995) for example has been shown to improve results for IR (Gonzalo et al., 1998). Another way to account for lexical variation is to index documents according to a domain Ontology (Hernandez, 2005). Using hand built substitution lists can also be considered as a symbolical approach. We discuss a symbolic approach in Section 4.1.2.
- Statistical approaches exploit co-occurrence patterns of words in large corpora. Also known as *distributional semantics* (Turney and Pantel, 2010; Baroni and Lenci, 2010; Padó and Lapata, 2007), such methods identify similarities between individual words based on the similarity of the contexts they appear in. Morlane-Hondère (2013) showed that such methods identify a large spectrum of semantic relationships. In employing them one can either produce substitution lists or they can

⁴We have encountered all the following variants of "checklist": check-list, check/list, c/list, c/l, c-l, checkliste

be directly incorporated in the vector space model, as is discussed in Section 4.3 and in Chapter 6.

In order to apply symbolic methods, to specialised texts as aviation occurrence reports, the main drawback is resource availability. Generic resources such as *Word Net* (Fellbaum, 1998) and general language thesauri are not adapted. *Ad-hoc* resource construction is necessary, which is an extremely costly process. The main obstacle for using statistical methods for with occurrence data is quantity. These methods require very large amounts of text and collecting such quantities in specialised domains is not a trivial task.

3.1.2.3 Compositionality and semantic variation

Semantic variation can be viewed as the inverse of lexical variation. A same word having different meaning. The textbook example of the “river bank” and “bank” as a financial institution is commonly used to illustrate semantical variation (Arampatzis et al., 2000). However, from an IR perspective the example of “altitude excursion” vs “runway excursion” illustrates the same phenomenon. The two cases are for our purposes equivalent, given the information need it is simply not the same kind of “bank” nor is it the same “excursion”. In order to treat this kind of phenomenon, one needs to take into account not the inherent meaning of words but their meaning *in context*. Or to put it in other words, the compositional character of meaning.

Compositionality is “the principle that the meaning of a (syntactically complex) whole is a function only of the meanings of its (syntactic) parts together with the manner in which these parts were combined.” (Pelletier, 1994).

A simple example is the support of polylexical units such as “runway excursion” from the example. Identifying such units can be done by symbolic means or in a pure statistical manner on surface forms. Symbolic methods rely on syntactic structure (Bourigault, 1993) and extract terms based on templates, such as looking for a sequence of nouns. Statistical methods can be as simple as extracting collocations (Kilgariff and Tugwell, 2001).

From an IR point of view this implies that indexing should account for structure. A simple way this can be done is by using *word n-grams* (contiguous sequences of words) as index terms. In more complex methods such as *term dependence models* (Croft et al., 2010), ranking is dependant not only on the frequency of the query terms but on their relative proximity in the documents.

In our experience with occurrence reports, we found out that considerable effort is needed for accurate syntactic parsing of the non-standard language common to aviation. For this reason for the time being we preferred a surface form based approach that we present in Section 4.1.5.

3.1.2.4 Discourse and document structure

Documents usually have structure (Power et al., 2003) and the different parts have different rhetorical roles (Mann and Thompson, 1988). Information present in different parts has different function and their relative importance differs.

From an IR perspective, taking into account discourse structure can be viewed as simply identifying the most interesting parts in a document and giving more weight to the terms found there.

A straightforward example is the status of titles and headings. In semi-structured documents, such elements are of greater relative importance than the text (Kronrod and Engel, 2001; Rebeyrolle et al., 2009).

Zones of text may also have different relative importance. In scientific articles, for example one would look for those parts that present new ideas and not the parts that expose previous work on the subject, as Teufel and Moens (2002) point before presenting a method to automatically distinguish such zones.

This was the main justification for Campello Rodrigues's Master's thesis (Campello Rodrigues, 2013). In it, we would explore ways of *zoning* accident reports (§2.1.2.2) and identify parts with different rhetorical roles, such as argumentative, descriptive and narrative.

3.1.3 IR for occurrence data

We saw in the previous chapter how occurrence data is organised and what the predominant uses are. Let us now see how this translates into the basic IR concepts we just saw.

3.1.3.1 Precise information needs

The need for information can be very precise. Users are trained and are experts in the field of aviation. Thus problem formulation can be highly detailed (see the ASRS example in 2.3.1). This is the main reason for the extensive use of taxonomies to organise the information in a well defined manner. However, as we saw, there is always a narrative part and full text search is often part of the query. In other words a well designed system must allow for a Boolean search over the metadata attributes as well as full-text search capabilities for the narrative fields.

3.1.3.2 Undefined information needs

Another scenario, also common is one where the information need is undefined or under-defined on purpose. In particular related to the need to look for patterns (§1.3 and §2.3.3), expressing and formulating an information need can be viewed as already introducing bias. To put it bluntly, one can not *look*

for the unexpected. This is the case when sifting through data on occurrences and just looking for “something out of the ordinary”. Thus a usage scenario of a purposefully designed IR system is also to allow easy browsing through the data, while providing the users with cues to potential patterns. The *timePlot* system, which we present in Chapter 5 was initially aimed at just such a usage scenario.

3.1.3.3 Favouring recall

Lastly a particularity we have repeatedly come through when interviewing safety experts, is their high tolerance to noise in the results coupled with the intolerance to silence. They did not want to miss anything out and are willing to accept a high error rate. In terms of parametrising a system, this means that a high recall (3.1.1.4) strategy is preferred. In terms of interface design, this means that a good IR system for occurrence data will facilitate identification of noise and isolation of the relevant results. In terms of NLP components, this means that recall-producing strategies, such as query expansion or dimensionality reduction could be favoured.

3.1.4 A broader perspective on the IR problem definition

Putting aside the common issues of IR and how NLP (potentially) solves some of them, it is also interesting to look at the broader problem of data-driven risk management from an IR perspective. Working on this thesis we have come to see that easy access to relevant information is central to the tasks at hand. As shown in the previous chapter (§ 2.3), experts always seek answers within large collections of semi-structured data. Whether by asking specific “questions” (queries), counting aggregations based on the metadata and plotting them on a bar chart or simply wondering if something interesting has happened lately, they express a need for information and are demanding tools that satisfy it. We need to relax the standard “user queries, system responds with a list of documents” paradigm just a bit and start asking again some of the fundamental questions of IR.

- **What is an information need?** The spectrum is large: From the simple navigational query, where a user searches for a specific document by its identifier on the one hand, through seeking instances of elaborate accident scenarios to seeking the unknown and assisting serendipity.
- **How is the user input formulated?** Is it a full text query, a combination of textual queries and metadata attributes, a question in natural language or a polygon drawn on a map?
- **How is the user involved throughout the search process?** Refining search through brief exploratory queries followed by more precise

and complex queries based on the initial results is common. What processing methods and what interface design choices can assist the users in understanding the data and the results presented?

- **Are the users trained or not to use one specific tool?** Do we assume that the user knows the underlying organisation of the data and the metadata model or do we need to abstract from them, assume that no such prior knowledge exists and present results in intuitive ways?
- **What does the user expect in return?** Is it a list of documents à la Google? a textual summary? a bar chart?

While not pretending to have a clear answer to these questions, asking them has helped us shed some light on the complexity of the domain at hand. While at the beginning of working on this thesis, we loosely framed our problem around automatic detection of certain scenarios, at the time of writing this document we are intimately convinced that rephrasing the problem as an information retrieval task is the key to designing and developing the future software solutions for risk management.

3.2 Automatic text categorisation

As we saw in the previous chapters (§2.2), taxonomies are the main means to access collections of incident and accident reports. They however suffer from a number of issues (§2.4.2), which all amount to either loss of coded data or lack of such in the first place. Deriving metadata attributes from the information contained in the natural language narratives is the task of automatic text categorisation.

3.2.1 Problem definition

The task of classification can be defined as “the task to classify a given data instance into a prespecified set of categories” (Feldman and Sanger, 2007). Applied to documents or texts, text categorisation (TC) is simply the process of finding the correct category (topic, theme, class, label) from a set of categories, for every document.

Automatic document classification is today part of our every day lives. The ads we see on web pages (Blaser et al., 2004) and the spam we don’t see in our inboxes (Cormack, 2007) are a result of TC techniques. Applications ranging from filtering potential job candidates for a job by scanning through their resumes (Singh et al., 2010) to detecting plagiarism in students’ writings (Lukashenko et al., 2007; Tanguy et al., 2011), from detecting subjective writing in news feeds (Kim and Hovy, 2006) or on *Twitter* (Pak and Paroubek, 2010) to trying to detect terrorists based on their email exchanges (Ahsan

et al., 2013) have made TC one of the more active fields in contemporary NLP.

TC is studied for at least 60 years (Maron, 1961) and there are basically two ways to attack the problem. The first is the *knowledge engineering* approach where expert knowledge is directly encoded in the system. In other words, manually produced rules or heuristics are applied to the documents and based on these rules the systems chooses one category or another. The other approach is the *machine learning* approach, where a classifier function is built inductively from a representative set of already classified examples. Although useful in some highly controlled environments, the knowledge engineering method requires considerable resources and expertise. This is why most of the research today is centred on machine learning methods.

The problem can be formally defined as approximating an assignment function $F : D \times C \rightarrow \{0, 1\}$, where D is the set of documents and C the set of classes. $F(d, c) = 1$ if one particular document d belongs to the category c . The *classifier* is the approximating function $M : D \times C \rightarrow \{0, 1\}$. The classifier needs to be as close as possible to the assignment function F (Feldman and Sanger, 2007).

Three aspects of statistical TC are important: the nature of the classification problem, the choice of classifier and how to represent documents.

3.2.1.1 The nature of the classification problem

Depending on the specific task we can distinguish between binary, single label and multilabel classification. Binary classification is the simplest task. The classifier needs to decide whether a document belongs to a given class or not. When there can be only one possible class for a document, the classifier needs to decide between all available classes and assign the most probable one. When each document can be assigned more than one class and the classes are not mutually exclusive, we have multilabel classification. Binary classification is the easiest task, single label classification is more difficult but can be solved by a generalisation of the binary case. Multi-label classification is more difficult but can be solved by applying as many binary classifiers as there are classes.

3.2.1.2 The choice of classifier

Statistical machine learning is a rapidly developing academic discipline and is by no means restricted to TC. New methods and algorithms are developed and released all the time and applied to all kinds of objects and scenarios. Machine vision for example, is highly dependant on statistical ML techniques. TC is just the application of general machine learning techniques to the problem of categorising texts. One has to choose between a number of different systems and often the choice depends on what is available and the current trend. Evaluating several different systems allows to make an informed choice but is

hard work. Often, when a given system has acceptable performance one sticks to it.

The different systems commonly used in TC applications are:

- Support Vector Machines. (Vapnik, 2006).
- Maximum entropy (Berger et al., 1996)
- Neural networks (or Perceptrons) (Bishop, 1995)
- Decision trees (Quinlan, 1986)

We will not discuss the different classifiers further. Our personal position on the subject is that any classifier that does the job well is a good choice. It is worth mentioning however that the models produced by some classifiers are less opaque than others. In some cases one might be interested in the exact reasons a given classifier produces a given outcome. Decision trees are by definitions easy to interpret as the model they build is a hierarchical structure of binary decisions based on a single feature. They are suited for more exploratory approaches. SVMs on the other hand are notoriously difficult to interpret and they are suited for result-oriented approaches.

3.2.1.3 Document representation

A classifier can not just take a text as input. One needs to transform the text in a set of features. We will discuss this in detail in the next chapter, but in a nutshell a document is represented by a *feature vector*. A feature is “simply an entity without internal structure - a dimension in feature space” (Feldman and Sanger, 2007). A document is represented as a vector in this space. The classification task can then be viewed as the process that assigns subspaces to the different classes.

3.2.2 Specifics of applying TC to occurrence data

Let us now see some of the challenges that may arise when using automatic TC methods for occurrence data. The issues that TC solves is incoherent or missing metadata (§2.4.2):

- Metadata incoherence, for example arises when a given coding standard changes and a collection starts accumulating reports coded by the new standard alongside the old one. In the original ADREP taxonomy, for example, the BIRD⁵ *Occurrence Category* didn’t exist and was added later. Originally bird-strikes were coded using the *ADRM*⁶ category. It is useful to recode the old reports in the new category.

⁵Bird-strike

⁶Aerodrome related issues

- Missing metadata is the case when new reports simply do not have coded values and the system assigns one to them automatically.

The first issue that arises with applying TC to this problem is the definition of the classification problem. There is only a tiny subset of all meta-data attributes that are classifiable in a straightforward manner. Rare are those attributes that are only simple lists of mutually exclusive values. The *flight phase*⁷ is such an example. More common are multilabel classification problems such as the ADREP *occurrence categories* (§2.2.3.5) whose classification we will discuss in the next section. Hierarchical categories are also common.

Another particularity is that, as metadata is organised in taxonomies, there is almost always some internal structure and domain specific logic and rules in its application. The *occurrence categories* are for example (in theory) divided in primary and secondary categories. A document *must* have one of 15 primary classes and *may* have one of 21 secondary categories. Thus the question is more complex than a multi-class TC problem (even though we treat it as such in the example in the next section). One more aspect of using complex taxonomies is the fact that there are more than one metadata attribute to classify and that informational redundancies can be established between the different branches. An aircraft can not experience an *abnormal runway contact*⁸ during *Approach*. Exploiting these links between different branches can potentially produce more coherent classifications.

Lastly and most importantly, the nature of the objects we classify are not documents in the strict sense. TC as it is defined is just that - assigning categories to texts or documents. With occurrence data however the object is not a text, it is a record of an event. Except with official accident reports (§2.1.2.2), text only vehicles some information about the occurrence. Other information is only present in metadata attributes. Thus mixed approaches need to be researched in order to combine the two complementary sources of information.

Data sparsity is another issue. Usually categories are unevenly distributed, as we will see in the next section.

For a mix of the above stated reasons, some parts of the metadata are simply too complex to be approached via TC techniques. With their high level of structuring, the *Events* of the ADREP taxonomy for example are a very hard problem for simple TC. Even if we look at them as a TC problem (that is that an *Event* is a label for the occurrence) the sheer number of values makes and sparsity of the data makes efficient categorisation highly improbable. The high-level of structuring (§2.2.3.5, Figure 2.24) however is coherent with a knowledge modelling approach.

⁷The *flight phase* simply denotes the different stages of a flight: standing, push-back, taxi, take-off, climb, cruise, approach, landing, taxi, standing.

⁸ARC occurrence category

3.2.3 Automatic classification of occurrence categories: an example

In this section we present work done in *Safety-Data*'s RD program and reported in (Tanguy et al., 2015). While not part of the main body of research in this thesis it is a perfect illustration of the applications of automatic document classification to occurrence data. This work was done in anticipation for replacing the original coding system used by *Safety-Data* (Hermann et al., 2008) which addresses the two scenarios we presented in the previous section (§3.2.2).

3.2.3.1 Context

As we discussed in the previous chapter, the DGAC is France's national aviation regulator and collects occurrence data from a variety of entities operating on French territory. Their database contains more than 400,000 occurrences collected over the past ten years, with approximately 45,000 incoming reports per year. Reports are mostly written in French (97%), although their authors make heavy use of technical aviation terms borrowed from English. The occurrences are coded with the ADREP taxonomy and stored using ECCAIRS (2.2.3.5). The branch of ADREP we were concerned with is the *occurrence category*. The full list of categories is available in Table 2.3.

Table 3.1 shows some of the categories with their associated descriptions and their relative frequency in the DGAC corpus.

Label	Description	% reports
ATM	Occurrences involving Air Traffic Management or communications, navigation, or surveillance (CNS) service issues.	40.6
BIRD	Birdstrike - Occurrences involving collisions / near collisions with birds.	7.3
RE	Runway excursion - A veer off or overrun off the runway surface.	0.7
GTOW	Glider towing related events.	0.03

Table 3.1: Examples of ADREP occurrence categories

3.2.3.2 Corpus size and category distribution

The database currently consists of 404,289 occurrence reports from 2004 until September 2014. Among these, only one third are labelled with at least one occurrence category. The corpus used in the study thus contains 136,861 documents, which amount to a total of 15 million words.

The categories themselves are very unevenly distributed as can be seen in the examples in table 3.1. The most common category is *ATM*, assigned to 40.6% of the corpus, while 25 of the 36 labels concern less than 1% of the reports. Some categories are very poorly represented: for example, *GTOW*, the category concerning glider-towing related incidents, concerns only 46 reports or 0.03% of the corpus (in addition to the rarity of these events, this category was recently added to the taxonomy).

The ADREP scheme considers that an occurrence can be described with more than one label, which leads to a multi-label classification situation. Among the labelled reports of the database, 95% have one category, 4% have two categories and only 1% have three or more (maximum 6).

3.2.3.3 Classifier and classification problem

We used the Support Vector Machines (or SVM) (Vapnik, 2006) supervised learning algorithm and used the features described in Section 4.1. Training (i.e. the construction of the predictive model) was performed with the java port of the *Liblinear* library⁹ (Ho and Lin, 2012).

As this is a multi-label classification problem (3.2.1.1), we trained 36 independent binary classifiers, one for each target category. This means that each report to be categorised is analysed by these 36 classifiers, and given an independent yes/no answer for its association with the 36 possible categories.

3.2.3.4 Results

Having a close look at the results, we found obvious inconsistencies in the original coding. One of the errors we identified was a common confusion between some of the categories and the *OTHR*¹⁰ category. When looking through the errors concerning the *RAMP*¹¹ category we identified that events concerning spillage of fuel while refuelling were (correctly) classified by the tool as *RAMP* events, while in the training corpus, roughly one out of five¹² such events had been attributed the *OTHR* category.

Table 3.2 shows detailed results of the classifier’s performance for various categories. It appears that our classifier gets very good results (with a precision exceeding 90%) for several categories, among which we can find some that are very frequent. For *ATM* and *BIRD*, both relatively frequent categories, the classifier performs well enough to allow for entirely automated classification with no human supervision.

Other categories are inherently difficult, even when frequently used. There are many components in an aircraft and they all may fail. The (non-powerplant)

⁹<http://liblinear.bwaldvogel.de/>

¹⁰Other - the catch-it-all category defined as “Any occurrence not covered under another category.”

¹¹Ground Handling - Occurrences during (or as a result of) ground handling operations.

¹²Determined by a manual examination of 200 documents.

Category	Count	P (%)	R (%)	F1 (%)
ATM	55614	96.31	93.09	94.67
BIRD	9943	96.08	93.01	94.51
MAC ¹³	7503	91.54	84.72	87.99
SCF-NP ¹⁴	9529	80.31	62.42	70.18
SCF-PP ¹⁵	2530	72.15	53.92	61.68
RE	943	87.62	77.61	82.04
GCOL ¹⁶	850	59.62	36.47	45.26

Table 3.2: Detailed scores per occurrence category

system component failure category *SCF-NP*, whose frequency is comparable to the bird-strike category, is much more difficult to recognise. The difficulty comes partly from the fact that a component failure will constitute a larger event and the crew’s actions (such as declaring an emergency, troubleshooting the error jointly with ATC) will be reported. This surplus of information creates a much harder problem to solve for the classifier.

Finally while data rarity is an obvious issue when considering machine learning approaches, it has not been too problematic in the present study. The *RE* category, for example, concerns only 94.3 occurrences on average and is classified with relative reliability. For other rare categories, such as *GCOL* (ground collision) the performance is much worse and can be attributed to a combination of rarity, difficulty¹⁷ and inconsistency¹⁸.

3.2.3.5 Industrialisation

In all these results validated the adoption of the system within *Safety-Data*’s commercialised tools. Given that performance varies according to the different categories, it was decided to adopt a hybrid strategy where certain documents will be coded in a fully automatic manner while for others the system will produce suggestions that should be validated by an expert.

A “high precision” strategy will be adopted for fully automatic classification. Given that each of the 36 binary classifiers produces a probability for a *yes* answer (that a given category describes a given document), we can calculate the threshold for which the system achieves a certain level of precision separately for each category. This level was set to 95%. If the probability for a given document and a given category is above the threshold, the category is automatically added. If it is below, the document is marked for manual validation, with the most probable categories presented in the form of suggestions to the user. For categories such as *BIRD*, where the system performs well, it

¹⁷There are several categories dealing with collisions.

¹⁸When reviewing the data, we are convinced that this particular category is largely under-represented: there are many events that should be coded *GCOL* and are not.

produces high recall, where for others such as RE, the recall is relatively poor. In both cases, though, the quality of the assigned codes is satisfactory.

This case is the inverse of the high recall strategy for IR (§3.1.3.3). While users tolerate a lot of noise in an IR context, where they have control and visibility of the results, in the classification task the importance is to have accurately coded data and the users both have a higher tolerance for silence and are willing to engage in manual validation of the borderline cases.

3.3 Chapter conclusion

In this chapter we presented the domains of Information Retrieval and Text Categorisation and how they potentially solve the issues presented by the uses of incident and accident data. Looking at the issues from an Information Retrieval perspective implies a range of considerations from defining the information needs, the documents to designing interfaces that support the complexity of the data the system handles. As both TC and IR require first processing the textual material present in the documents, in the next chapter we will discuss how text is transformed into an input material for IR and TC applications.

Chapter Four

From text to vectors

*“My dark and cloudy words they do but hold
The Truth, as Cabinets inclose the Gold.”*

— John Bunyan, *Pilgrim’s Progress*

This chapter is divided in two parts. First, in Section (4.1) we present our solution to the problem of normalising the textual material we encounter in incident and accident reports in order to transform it to formats suitable for vector space modelling. Next, in Sections 4.2 and 4.3 we discuss the vector space modelling framework and present the notion of dimensionality reduction, central to many current NLP methods.

We saw in the previous chapter two NLP applications, Information Retrieval for searching large document collections and Text Categorisation, for classifying documents within taxonomies. In order to be able to access the information contained in the texts, both these applications require first a transformation of the raw material into representations suitable for numerical processing. This chapter will describe this transformation.

The predominant paradigm for processing language today is Vector Space Modelling (Salton et al., 1975; Turney and Pantel, 2010). Documents or texts are represented as points in a high dimensional space. The process has two parts - a symbolic transformation, where the texts are broken down into discrete descriptors - the features - and a numerical transformation where, based upon these features the texts are transformed into mathematical objects - points in a n -dimensional space. This step is essential for many NLP applications in general and for all the applications we discussed in the previous chapter. A search engine (§3.1) for example needs to be able to compare a query to a document. This comparison is done on the base of shared features between the two. Likewise in order to find similar documents, the system will search for those that share the most features. Text Categorisation will assign a subspace of the vector space to each category.

The processing chain and the principles we present here are ongoing work in *Safety Data* and stem from the will to factorise all the linguistic processing in a single processing chain. Rather than build separate systems for separate tasks, the feature extractor we describe here is used to process documents for the full-text search engine, for calculating similarity between texts for the text categorisation applications such as the one presented in Section 3.2.3.

4.1 Extracting features

Feature extraction is the process of computing a set of discrete descriptors. A feature is a key-value pair where the value is either of a boolean type or numerical, “an entity without structure”, as Feldman and Sanger (2007) put it, “a dimension in feature space”. The sum of these descriptors are known as the feature set. We can distinguish low-level features, which are features that are extracted *from* the document and high-level features, which are features computed *for* a document (usually based on low-level features) or attributed to the document based on an external resource (such as metadata). A low-level feature will be an expression stating that “this text contains the word *bird*”. A high level feature would be the expression stating that “this document has the value *BIRD* as its *occurrence category* attribute”. We will now discuss extracting low-level features from text and the challenges this process presents.

Given that we are after the meaning of the text, the first goal of feature extraction is to keep as much of the information present in the texts as possible. Meaning is conveyed by the words so they are the obvious candidates. The

second goal of feature extraction is to account for as much surface variation as possible. Words, as we saw (§3.1.2) do not always appear in exactly the same form. We project linguistic knowledge about the modes of variation in order to capture such surface equivalences. Also, feature extraction should concern itself with the inherent noise in the surface forms. not all features are vehicles of meaning. The word “the”, in isolation does not inform us in any way about the meaning of the text it comes from. The words “landing” and “gear” taken separately have their inherent meaning but it is not as precise as the meaning of the compound “landing gear”.


4.1.1 Tokenising

The first step of any feature extractor is splitting the text up into individual tokens - the words (Grefenstette and Tapanainen, 1994). In western languages such as English this step seems trivial and often a basic whitespace *tokeniser* does the job to a very satisfying degree. But even for English a tokeniser must be able to handle ambiguous punctuation such as hyphens, full stops in acronyms and other borderline cases. For other languages, however tokenisation is much more difficult. Chinese for example does not use whitespace to separate words, so even at this basic level, a sophistication, such as dictionary matching of sequences is required.

4.1.2 Levels of normalisation

When text and language are concerned variation is omnipresent. Whether presented with two words that have the same meaning or two competing translations of the same book, humans have no particular problem at constructing the correct mental representation of the information they vehicle. For computers even the slightest difference is significant. Such differences should be accounted for by a successful NLP system, which in a way should be able to tell when to consider two forms equivalent and when to consider them different. In the previous chapter (§3.1.2) we saw how variation impacts IR. Here we will take a closer look at our process for handling some of the aspects of linguistic variation.

First of all, variation is recursive. We usually present the different levels separately, but as the example from the ASRS database illustrates an overlap exists between them. In it, variation occurs on both the level of individual terms and in the manner in which they are combined. The concept of “maximum takeoff weight” can also be referred to by an acronym, “MTOW”. However components of the developed form, “weight” and “maximum” can be abbreviated in “wt” and “max”. The term “takeoff” can be spelled in several ways. All this variation produces a great number of combinations as illustrated in figure 4.1, where all thirteen terms are strictly equivalent. Their multitude is generated by an articulation of variation on three levels:



MTOW
 MAX TKOF WT
 MAX TKOF WEIGHT
 MAX TAKE OFF WT
 max takeoff weight
 max take off weight
 Maximum Take Off Weight
 Maximum Take-off Weight
 maximum take-off weight
 MAXIMUM TAKEOFF WEIGHT
 Maximum Takeoff Weight
 Maximum takeoff weight
 maximum takeoff weight

Figure 4.1: Variants of MTOW

- **Character level:** Variation between upper and lower case needs to be taken into account. It might seem trivial to normalise, but in many cases even character variation can be problematic. Characters with diacritics in French, can sometimes be written deaccentuated, especially when the whole word is in uppercase. Choosing to normalise accents into their deaccentuated variant will inevitably bind some words with different meanings, such as “côte” (rib, coast) and “côté” (side) to a single feature. Choosing not to will produce unrelated features where the same word is considered if one of the inputs is capitalised. Case variation also plays a part in the treatment of acronyms, such as ILS¹ which should be differentiated from the french third person pronoun “ils”.
- **Token level:** Token level variation concerns equivalent forms with different spellings. In the example, the word “weight” and its abbreviated variant “WT” must produce the same feature. More commonly, morphological variants such as verb inflexions and plurals need to be folded down to a single form. Pushing the limit one might even consider synonymy and word-to-word translations at this level.
- **Supratoken variation:** Equivalence between forms of different grain size such as variants of the term “takeoff” and short and developed forms of acronyms are examples of such variation, showing the limits of considering words as atomic elementary units. Without such considerations, one will produce noise in the form of features such as “take”, “off” or “take-off” (depending on how one considers the hyphen). Considering it implies using resources (and their construction if they are unavailable) and substitutions on a case by case basis. Supratoken variation is also

¹Instrument Landing System

addressed by any normalisation, performed on the syntactic level such as for example transforming passive into active voice.

The example in figure 4.1 shows how several levels of variation articulate and produce different surface forms. A feature extractor should be able to take into account as much as possible such types of variation. We will now see a processing chain built to account for such cases.

4.1.3 Overview of a processing chain

Figure 4.2 depicts the processing used by of *Safety Data*'s² linguistic processing module, in the state a few months prior to the time of writing this thesis. It is responsible for the feature extraction for all the NLP - related tasks dealing with incident and accident reports, namely automatic report classification (§3.2), similarity calculation (which will be discussed in detail in §5.1) and a full-text search engine (§3.1). The different applications do not however all use the full set of features.

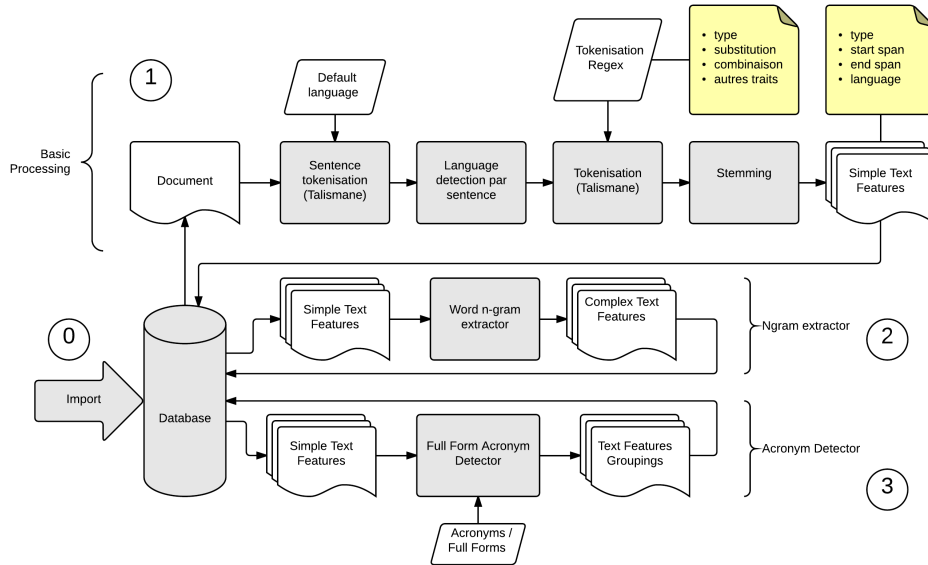


Figure 4.2: A processing chain

²We present it here to discuss and exemplify some of the design choices that deal with the problems presented in the previous section. While we have participated in some of the said choices we do not by any means claim ownership of this work, which is a joint effort by the *Safety Data* team.

The example in figure 4.3 is the synopsis of a report from the ASRS database. It is written in an abbreviated concise writing style, typical for ASRS until a few years ago. It contains acronyms, such as NMAC and abbreviations such as “RWY” (runway). We will use it to exemplify the different steps of the process.

Table 4.1 shows the end result - the features extracted by the processing chain.

A SMA ^a LANDED WITHOUT CLRNC AS ANOTHER ACFT WAS TAK- ING THE RWY CREATING A NMAC ^b
^a Small Aircraft ^b Near Mid Air Collision

Figure 4.3: Synopsis of ASRS ASN796443

There are several consequential stages of processing:

- **0 - Import:** The preliminary stage dealing with extraction of the text data and storing it in the appropriate record in a relational database. We will not discuss this stage in details. It is sufficient to say that at the end narrative parts are extracted from whatever format they originally arrive in (html, xml, excel, relational databases, ECCAIRS and various proprietary formats).
- **1 - Basic processing:** This stage is common to all the applications and produces the basic features that describe each document. All subsequent stages are based upon the results of this stage.
- **2 - Word n-gram extractor:** This stage produces complex features (word n-grams).
- **3 - Acronym Detector:** This stage produces features responsible for indexing domain specific acronyms.

We will now see in grater detail the different stages of processing that produce this output.

4.1.4 Basic processing

Basic processing takes a text as input and produces a set of word stems as output. The first stage is to determine sentence boundaries. This is done by a statistical sentence *tokeniser*, part of the Talismane (Urieli, 2013) parser. Next, for each sentence a language detector determines the language of the sentence. This might seem unorthodox, but some texts may have mixed languages (see Figure 2.14 for an extreme example of such a report). For this reason, very early on in the processing, language detection on the sentence

Id	Type	Name	Span Start	Span End
1	<i>stopWord</i>	a	0	1
2	<i>acronym</i>	SMA	2	5
3	<i>wordNgram</i>	SMA land	2	12
4	<i>word</i>	land	6	12
5	<i>acronymExpansion</i>	LWOC	6	26
6	<i>wordNgram</i>	land without clearanc	6	26
7	<i>stopWord</i>	without	13	20
8	<i>word</i>	clearanc	21	26
9	<i>wordNgram</i>	clearanc as anoth	21	37
10	<i>stopWord</i>	as	27	29
11	<i>word</i>	anoth	30	37
12	<i>wordNgram</i>	anoth aircraft	30	42
13	<i>word</i>	aircraft	38	42
14	<i>wordNgram</i>	aircraft was take	38	53
15	<i>stopWord</i>	was	43	46
16	<i>word</i>	take	47	53
17	<i>wordNgram</i>	take the runway	47	61
18	<i>stopWord</i>	the	54	57
19	<i>word</i>	runway	58	61
20	<i>wordNgram</i>	runway creat	58	70
21	<i>word</i>	creat	62	70
22	<i>wordNgram</i>	creat a NMAC	62	77
23	<i>stopWord</i>	a	71	72
24	<i>acronym</i>	NMAC	73	77
25	<i>punctuation</i>	.	77	78

Table 4.1: Features extracted from ASRS ASN796443

level is needed and information about the language is passed on at the latter stages.

Next, each sentence is *tokenised* by a regular expression based tokeniser. The sentences are split up into individual words, essentially corresponding to the features of types *word*, *stopWord*, *acronym* and *punctuation* in table 4.1.

At this stage pre-processing and normalisation rules are applied to known variants and abbreviations. For example the strings “CLRNC”, “ACFT” and “RWY” in the original text are replaced by their expanded variants - “clearance”, “aircraft” and “runway”.

Also, regular expression based detection of special types of tokens, such as dates, urls or units of measurement as well as the stopwords. The latter are kept in the feature list with a dedicated type as they are useful for later stages of processing.

Next, a *stemmer* reduces morphological variants to a common root. Stemming is done by the *Snowball*³ stemmer. The word “landed” becomes the feature “land”. Other morphological variants, such as “landing” and “lands” will also be reduced to the stem “land”.

At this stage the basic processing is complete. Positional information in the original text is kept, as can be seen in table 4.1. This is necessary in order to keep the link with the original text and be able to display it to the user in the form of highlighted text. The features are stored in the database.

4.1.5 Word n-gram extractor

The second stage takes the simple features extracted by the basic processing and combines them in multi-word strings. The word n-gram extractor uses simple rules and extracts every contiguous string of n stems that does not start or end with a stopword and does not contain punctuation or a sentence boundary. In table 4.1, word n-grams correspond to features with the type *wordNgram*. The extractor is configured to extract n-grams of length 2 and 3.

Multi-word tokens are useful as they capture more specific information than stems alone. For applications such as automatic document classification, they contribute to a better precision of the classifier, as we show in (Tanguy et al., 2015).

The simple n-gram approach was chosen for its robustness and its ease of implementation. While less sophisticated than full scale syntactic analysis or even *chunking*, the word n-gram approach produces features of the same nature. The underlying rationale behind this design choice is to keep it simple at first, observe the behaviour of such features, evaluate their usefulness and decide *if* and *how* more sophisticated processing should be applied and for what tasks.

4.1.6 Detector of developed acronyms

The third stage is a detector of developed forms for acronyms. Taking simple text features as input and a long list of domain-specific acronyms associated with their developed forms, the module looks for contiguous stems that form the developed form of an acronym. When it finds such a configuration it creates a new feature in the database, having the acronym as its label and pointing to the actual positions of the developed form in the text.

In table 4.1, feature number 5 represents the developed form of the acronym “LWOC”(landed without clearance). We can see how the positional information (spans) reflects that it concerns the full twenty characters of the developed form. In this way, texts containing both forms are indexed in a coherent manner. Other documents containing the acronym “LWOC” in its condensed

³<http://snowball.tartarus.org/>

form will be assimilated with this document. Likewise, documents containing a developed form of “NMAC” or “SMA” will be assimilated with it as well.

This step ends the feature extraction process.

4.2 Representing documents in vector space

We will now see the next step of the transformation of a text into a vector.

When we process incident and accident reports, each *occurrence* manifests itself as a document stored in a particular collection. We want to be able to manipulate these documents, discriminate between them, compare them one to another, classify them in accordance to some schema or another or identify those that contain new and unseen information. In other words we want our functions to have access to the *meaning(s)* of the individual documents and to manipulate their meanings relatively to one another.

But computers and meaning don’t get along very well and for a very good reason. Meaning (and humans) lives in a universe of qualities, where symbols, concepts, definitions, characteristics and metaphors are manipulated to create it. Computers exist in a universe of quantity and manipulate only numbers. While in the previous chapter we saw how to *represent* qualitative material for computers to manipulate, there is still one major hurdle to overcome before they can actually start *using* it to produce new knowledge: we need to transform it into quantities, into mathematical objects.

Vector Space Models (VSMs) are a simple and elegant bridge between these two (incompatible) universes.

VSM models for document collections have been around since the 1970s and were introduced in the SMART IR⁴ system (Salton et al., 1975) and today are arguably the most widespread approach to semantics. Today they are extensively used both in industry and in academic research. In industry most (if not all) search engines rely at some point on VSMs to match the user query against the documents in the collection. A growing body of work ranging from cognitive psychology and sociology through almost all sub-disciplines of linguistics have used VSMs with large document collections either to analyse the structure and content of the collections or to gain knowledge on different aspects of the functioning of language itself (Turney and Pantel, 2010).

The general idea of VSMs is to represent an individual of a given collection as a point in a space (or a vector in a vector-space). The relatedness (similarity) of the individuals is proportional to the distance between the points representing them. Point that are close together represent related individuals and point that are further away - unrelated individuals.

It is convenient to organise the collection in a matrix, where the rows correspond to the individual documents, each one represented by a document-

⁴Information Retrieval

vector and the columns to a set of *features* that describe them. In the simplest possible VSM for representing a collection of documents, the features will correspond to the words contained in the documents. Each document will be represented by a *bag* (or *multiset*⁵). For example, for the document “Every day is a new day” the corresponding bag will be $\{a, day, day, every, is, new\}$. We can represent the bag with the vector $\mathbf{x} = \langle 1, 2, 1, 1, 1 \rangle$, where the first element in the vector is the frequency of *a* in the bag, the second element the *frequency* of *day* and so on. Thus the collection of documents (or the set of bags) can be represented as a matrix \mathbf{X} where each row \mathbf{x}_i corresponds to a bag, each column $\mathbf{x}_{:j}$ to a unique member and each element x_{ij} to the frequency of the *j*-th element in the *i*-th bag.

4.2.1 The term matrix

Building the term matrix is straightforward. It consists in sequentially scanning the texts, extracting the features with a given feature extractor (such as the processing chain we saw in the previous section) and then building a sparse matrix representation.

[DOC 1] A SMA LANDED WITHOUT CLRNC AS ANOTHER ACFT WAS TAKING THE RWY CREATING A NMAC.

[DOC 2] A PA28 LANDS ON THE TXWY INSTEAD OF THE RWY.

[DOC 3] SMT X VISUAL APCH TO WRONG RWY HAD NMAC WITH SMA Y. SEE AND AVOID CONCEPT.

Figure 4.4: Synopses of ASRS ASN796443, ASN356951 and ASN241893

Table 4.2 shows the term matrix constructed after analysing the three documents in figure 4.4. For convenience we have only represented single token features. The term space constructed from these three examples corresponds to the vocabulary of the (tiny) corpus and after removing *stopwords*. It is composed of twenty four unique features. Thus the space is said to have twenty four *dimensions*. Each document is represented by a vector in the space and its coordinates on the *n*-th dimension correspond to the frequency of *n*-th feature in the document.

4.2.2 Feature weighing

Weighing is a statistical transformation of the matrix in order to better capture the informational content of individual features based on their frequency. Word distribution in language follows an exponential curve. A few words are very common and thus very likely to appear in texts. A lot of words only

⁵A *set* where duplicates are allowed

Id	Type	Name	Doc 1	Doc 2	Doc 3
1	<i>word</i>	aircraft	1	0	0
2	<i>word</i>	anoth	1	0	0
3	<i>word</i>	approach	0	0	1
4	<i>word</i>	avoid	0	0	1
5	<i>word</i>	clearanc	1	0	0
6	<i>word</i>	concept	0	0	1
7	<i>word</i>	creat	1	0	0
8	<i>word</i>	had	0	0	1
9	<i>word</i>	instead	0	1	0
10	<i>word</i>	land	1	1	0
11	<i>acronymExpansion</i>	LWOC	1	0	0
12	<i>acronym</i>	NMAC	1	0	1
13	<i>brand</i>	PA28	0	1	0
14	<i>word</i>	runway	1	1	1
15	<i>word</i>	see	0	0	1
16	<i>acronym</i>	SMA	1	0	1
17	<i>acronym</i>	SMT	0	0	1
18	<i>word</i>	take	1	0	0
19	<i>word</i>	taxiway	0	1	0
20	<i>word</i>	visual	0	0	1
21	<i>word</i>	with	0	0	1
22	<i>word</i>	wrong	0	0	1
23	<i>word</i>	x	0	0	1
24	<i>word</i>	y	0	0	1

Table 4.2: Features extracted from ASRS ASN796443

rarely appear in texts. If we think about features as *events* in an information-theoretical perspective (Shannon, 1948), then a surprising *event* has more informational content than an expected event. In this sense, *weighing* gives higher numerical values to surprising events than expected ones.

The most common way to formalise this in the context of term-document matrices is the *tf-idf* family weighing functions (Spärck-Jones, 2004). Weight is computed as the product of the frequency of a given term in a given document (*tf*) and the logarithm of the inverse frequency of a term in the collection (*idf*). The more frequent a term in a document, the more it is informative for that particular document. However the more frequent a term is in the collection, the *less* important its overall informativeness is.

Consider the terms “land” and “NMAC” and consider the feature matrix in table 4.2 as part of the corresponding matrix for the whole corpus of ASRS reports. The two terms appear once in documents 1 and 3. Given that the two documents are of about equal size, the two terms are equally informative in the

two documents. “NMAC” however appears in (only) 6761 of the total 339320 texts in the collection, whereas “land” is present in 98609 texts. Their *idf* values are then respectively 1.7 and 0.53, reflecting the relative importance of each term. When considering similarity, for example the fact that two documents share “NMAC” will be considered about 3.5 times more important than if they share “land”.

Tf-idf is far from the only weighing function out there. Many exist, such as *PMI* (Turney, 2001) or *Okapi/BM25* (Spärck Jones et al., 2000) and, as *tf-idf* itself, each has countless variants.

4.3 Dimensionality reduction methods

Operating on a term document matrix still has one major drawback. The *orthogonality* of dimensions echoes the discreteness of the features. In other words the mathematical object constructed is still very closely related to the concrete surface forms (minus whatever linguistic normalisations are performed by the feature extractor) and rather far away from the (abstract) “meaning” that all text processing applications are ultimately after and for which text is but a vehicle and which is constructed progressively by combining terms in larger structures. In order to move still further away from the surface representations, a number of techniques have emerged, that essentially transform the initial high-dimensional term-space into a more concise space with less dimensions, providing just such a more abstract way of representing texts as vectors.

4.3.1 Smoothing the term matrix

Without doubt the most famous dimensionality reduction method for text is *Latent Semantic Analysis* (LSA) (Deerwester et al., 1990) and its related LSI (Hofmann, 1999b) and PLSA (Hofmann, 1999a). LSA consists compressing an initial term matrix by a mathematical operation known as a singular value decomposition (SVD). SVD essentially takes a (large) matrix as input and produces one of (much) smaller dimensionality, approximating at best the original matrix.

Turney and Pantel (2010) looks at the effects of LSA in four ways:

- **Latent meaning** As put forward by the authors of the method (Deerwester et al., 1990), compressing a matrix with SVD can be seen as a method of discovering latent meaning. The low dimensional mapping of a high dimensional term space captures the relationships between the terms and their contexts and forces and words with similar meaning end up mapped to a single dimension, representing their “collective” meaning.

- **Noise reduction** Thinking of the original matrix as composed of signal and noise, compressing it (reducing randomness) tends to capture variation due before all to the signal.
- **Higher order co-occurrence.** While words appearing in *identical* context can be said to have similar meanings, LSA captures also the relationship between words appearing in *similar* contexts.
- **Sparsity reduction.** Reducing the dimensionality of the original matrix from several tens of thousands to several hundred dimensions has the advantage of providing an object that is much easier for machines to manipulate.

The first two are the most important and are highly related. The process of compressing the original matrix grounds itself upon the inherent redundancy of surface forms found in text. When this redundancy is predictable within a collection there is a high probability that the surface forms are related. If many texts in the collection contain the words “bird”, “strike”, “collision” but also “seagull”, “goose” and “falcon”, the algorithm will determine that these terms can be mapped on a single dimension (instead of 4) without any major loss of information. This dimension can be then interpreted as related to bird-strikes, hence the notion of latent meaning.

Since LSA, many other methods of matrix compression have been invented, most notably *Topic Modelling* Blei (2012), which we have applied and tested on the ASRS database (see §6.4).

It is interesting to note that comparable results can be achieved without complex processing of the term matrix as a whole. Random Indexing (Sahlgren, 2005) is a dimensionality reduction technique that does nothing more than represent the individual features as vectors that are the sum of their contexts. By initially assigning a sparse random vector to each feature and then iteratively scanning the texts and summing the vectors of the features in its immediate vicinity, this method achieves a similar more abstract representation of word meaning.

4.3.2 Explicit methods

Another way to look at the problem of sparseness, noisiness and concreteness of the vector space is by determining beforehand a vector space with the desired level of abstraction and then finding a way to index a collection within that space. This is exactly what methods such as *Explicit Semantic Analysis* (ESA) (Gabrilovich and Markovitch, 2007) provide. The idea behind ESA is that one can use an external resource (Wikipedia in the original implementation) with a more “conceptual” structure. Individual articles in the online encyclopedia can be viewed as representing real-world concepts. There exists, for example an article on bird-strikes and, as one would expect it contains a subsection called “Species”, listing the most common unfortunate aviaries which cause damage

to aircraft. So, instead of looking to establish relatedness intrinsically, based on term co-occurrence within a large matrix, one could exploit the content of the Wikipedia article to establish these links. Thus, in ESA, the resource is used as an intermediate mapping layer between a high-dimensional term-space and a lower-dimensional concept space provided by the individual articles.

We explain the process in greater detail in section 6.3, where using ESA with a purposefully crafted multilingual resource, we present a system allowing for language independent indexing of documents in English and French.

Explicit methods can be viewed as a somewhat top-down approach to constructing the vector-space. In fact, Gabrilovich and Markovitch (2007) lengthily explore different ways to establish a right grain size for the final vector space by exploiting the hierarchical organisation of Wikipedia. Essentially, by concatenating articles grouped by a common class at a certain level of the encyclopedia’s categorical hierarchy, they provide spaces with different levels of abstractness.

Explicit methods for us are of particular interest as within the highly specialised domain of aviation safety, conceptual structure and organisation is already provided by coding taxonomies and meta-data attributes. The question is what is the relationship between the information contained in the report narratives related to their metadata attributes, and how to exploit taxonomy structure in order to provide more realistic and “conceptual” vector space mappings. In section 6.2 we present a preliminary approach to the latter question, where we use metadata to filter out dimensions that we already have information about via the coded attributes. In section 6.4 we show that there is significant overlap between the *topics* (dimensionality reduction mappings) produced by *Topic Modelling* and the taxonomy structure of the ASRS database.

4.3.3 Intrinsic or extrinsic, hidden or explicit?

Common to all dimensionality reduction methods is that they seek to define a function that maps a sparse, noisy and concrete vector space to a more concise, less noisy and more abstract one which is (in theory) closer to the true nature of the meaning conveyed by texts and further away from its variable surface manifestation. There are two ways to look at the problem. First, how do we define the space. Does it emerge from the collection itself, or do we use some kind of external resource to obtain it? Second, how interpretable the resulting dimensions are and can we operate on them on a dimension par dimension basis?

Method such as LSA and Topic Modelling define a space based on the collection alone. Yet, in reality all these methods first “learn” the mapping and then apply it. So it is possible to train a model on an external resource and then apply it to the collection we are interested in indexing. In fact, LSA distributes such a mapping learned on a corpus deemed large enough by

the method’s authors and providing “good enough” latent dimensions, so that an interested party can index even small collection without having to worry about first constructing the model. Conversely, the “explicit” nature of ESA is not mandatory. Claveau (2012), for example, shows that one can use ESA-like methods in an intrinsic fashion, by constructing a second-order mapping using the documents from the indexed collection itself.

The second point of interest is the interpretability of the reduced dimensions. What ESA prides itself upon is that the dimensions of the resulting reduced space are directly interpretable (hence the E for “explicit”). One can at a glance determine the *reason* that two texts are considered similar by a system by looking at the titles of the Wikipedia articles they are associated with. LSA avoids altogether the question by stressing on the “latent” and “hidden” nature of the resulting dimensions. The rhetoric around Topic Modelling is more nuanced. It puts forward the interpretability of sets of associated terms that form the dimensions. As one can see by looking at the examples in section 6.4, these sets of terms are surely coherent, but nonetheless only provide a very basic insight into the nature of the resulting dimensions and require a great deal of interpretive effort to be usable in a real world⁶ indexing scenario. Furthermore, in our opinion, where interpretability is concerned, there is no fundamental difference between *Topic Modelling* and the other smoothing methods, such as LSA where one could extract from the model the n terms with the highest loading for any given dimension.

4.4 Chapter conclusion

In this chapter we saw how text is transformed into vectors of increasing levels of abstraction. We saw a concrete example feature extraction and its adaptation to the domain in the form of specific levels of processing. We also saw the mathematical transformation that a feature set undergoes in order to be modelled in a vector space, as well as different dimensionality reduction techniques, aimed at attaining even higher levels of abstraction.

In the next chapter we will see how a simple vector space model is used as the base for an application destined at identifying similar incident and accident reports.

⁶If we want to provide the reason for a given similarity score to a user, for example, it would be quite cumbersome to show him several columns of related terms.

Chapter Five

The *timePlot* system: detecting similar reports over time

One cannot hope thus to equal the speed and flexibility with which the mind follows an associative trail, but it should be possible to beat the mind decisively in regard to the permanence and clarity of the items resurrected from storage.

— Vannevar Bush *As We May Think*

In this chapter we present the *timePlot* system we have built for detecting similar occurrence reports. Section 5.1 presents the problem and how similarity between documents is computed. In Sections 5.2 and 5.3 we present the tool's graphical interface and example of the results it presents to the users. Finally, in Sections 5.4 and 5.5 we discuss how the tool was used and discuss how, by observing the users' interactions with it, we came to gain further insight into their actual needs.

As we saw in Sections 1.3 and 2.3.3, one of the main challenges, when working with databases of occurrence reports, is the identification of recurrent risks. An obvious manifestation of such risks are multiple distinct events with almost identical or very similar circumstances. When these events are subject to incident or accident reports, it is possible to identify them by comparing the individual entries in a given database and representing their resemblance by computing a *similarity score* for each pair of entries.

In a way most events recur all the time. It is not because an event recurs that it is automatically of interest. However if a certain type of event starts recurring more than usual, it might indicate a pattern. For this reason the tool we present combines the notion of *similarity* and the chronological distribution of similar events - the “time” in *timePlot*¹.

Now we will present how we apply the existing methods of calculating textual similarity to the specific task at hand and what the particularities of the data and the way it is used can teach us about the different methods we tested and envisioned. Given the heterogeneous and unstable nature of the coded data (§2.2.2, §2.4), the initial focus was to build a “similarity analysis” system using only the narrative data as a source. Such a system has the benefit to be robust and uninfluenced by the many issues and biases of the coded data. Also, by explicitly considering only the textual parts of the reports the scope of such an analysis is extended to databases with little or no coding and without a clearly defined taxonomy. Such is the case of “young”, undefined, or constantly evolving reporting architectures (§2.4.2.3). Loss of coded data may also occur due to *bottleneck effects* when data is exchanged between institutions using different standards and formats (§2.4.2.4).

As we saw in Section 4.2, geometrically representing text is one of the fundamental methods in modern NLP, bridging the gap between the inherently symbolic nature of human language and the numerical objects that machines manipulate with ease. Both robust and simple to conceive and maintain, vector-space modelling was the method we chose for building the system.

The basic idea was to exploit the narrative parts of incident and accident reports and, by computing a similarity score between each pair of documents in the collections to generate a layer of structure that is presented to the user in the form of an interactive visualisation.

From an end-user’s perspective, we intended to develop a system requiring a minimum of initial while allowing interactive browsing of a database of incidents. The basic idea was to stimulate the expert’s serendipity by explicitating sets of occurrences, that might indicate a pattern. For this an interactive visualisation replaced the more traditional list of results we are accustomed to find. This echoes the unspecified information need we described in Section

¹We kept the name of the very first prototype. “timePlot.pl” was the file name of the perl script which *printed* an outrageous *html* file with data for the similarities pre-loaded in *javascript* variables.

3.1.3.2.

From a system's perspective we willingly kept the things simple by only focusing on the texts in the reports narratives. This allowed both to be immune to the metadata-related issues discussed in Section 2.4.2 and to use the tool as a basis for exploring how the narratives relate one to another and effectively use the system as a stepping stone towards the more complex methods discussed in Chapter 6. In hindsight this choice proved a wise one, as the system was rapidly proposed as a service by *Safety Data* and its simplicity allowed it to scale to databases of close to half a million documents as well as to be deployed for clients from other fields than aviation with little or no metadata.

In the next sections we will first discuss how we compute similarity between documents, next we will present the user interface in detail and we will show several incident scenarios with different chronological distributions identified by the system. Last, we will examine how the system was deployed at various institutions and the lessons learned from examining how it was really used.

5.1 Calculating similarity

5.1.1 At the prototype stage

The core of the similarity calculation at the prototype stage was straightforward. Given any pair of documents, the system produces a similarity score, between 0 and 1 representing the relatedness of the documents. The score is based on the *lexical overlap* of the narrative parts of the two documents. The more words they share in common the more similar the documents are. This is a classical implementation of the vector-space Information Retrieval principles (Manning et al., 2008) (§3.1), only that similarity is calculated between documents and not between a query and a document.

For extracting the features, we used the *TreeTagger* POS tagger (Schmid, 1994) with the stock model provided by the package.

We removed terms based on their POS-tags, keeping only nouns, adjectives and verbs. In order to normalise morphological variation we used the lemmas provided by the the tagger. In the (very numerous) cases where no lemma was produced we used the surface form.

Each document was then represented by a vector where each dimension corresponds to a term in the collection, and each value is the relative weight of this term in the document (§4.2).

We used the classical TF*IDF score (Spärck-Jones, 2004) to weigh (§4.2.2) the document vectors.

Finally for the similarity score between two documents we calculated the cosine (or dot product) between the vectors that represent them.

The processing was done by *Perl* scripts in an offline mode, on a static collection and produced in the end a square similarity matrix containing the

similarity score for each pair of documents. This matrix was then pruned for computational efficiency, discarding all the scores below a fixed threshold (0.10), removing more than 90% of the similarity scores, before manually loading the results in the system for visualisation and browsing.

5.1.2 From the prototype to a functioning system

When first shown to *CFH-Safety Data's* clients the initial prototype was very well received, as the “success stories” we present in the next section (5.4) show. It was deployed both at the DGAC and at a large national airline. The need quickly emerged for the system to be able to handle large collections of documents, coming from dynamic collections (frequently updated). While validating completely the similarity based approach for detecting patterns, this need made us completely rethink the technical details of how the score is computed. The static system from the prototype was incompatible with these new requirements. So we switched to the indexing functionalities provided by the *Lucene* search engine, effectively integrating it as the system's back-end replacing the offline similarity calculation by a on-the fly calculation. We used the built-in *stemmer* to process the texts, the stop-list provided by the package² and kept the stock configuration of the *Lucene* package.

In place of calculating similarity in the way described above, we treated documents as queries, feeding the text of the document's text to the system as if it was a query provided by the user. The system identifies the documents best matching the query and returns them providing a score reflecting how well the document matches the query. We consider this score as equivalent to the similarity score, normalise it to be between 0 and 1, apply a threshold and return the documents to the user.

Adopting the *Lucene* search engine as a back-end also solved the technical problems of updating the database incrementally (new documents can be indexed on the fly) and allowed the possibility to provide full-text search capabilities, which became one of the most used features of the system. *Lucene* also provided the much appreciated functionality to construct Boolean queries on both the text and the metadata attributes.

We will now see how the results are presented to the user and how he interacts with the system.

5.2 Presentation of the tool

In this section we will present the tool. The examples come from the DGAC's database, containing just over 400,000 reports at the time we took the screenshots.

²We use *Lucy*, a Perl port of *Lucene* (<http://search.cpan.org/~creamyg/Lucy-0.4.2/>)

The interactions start when the user selects a report that he is interested in. This is the *source report*, to which all the other reports in the database will be compared. The user has several choices for selecting the source report: He can either use the search engine to query the meta-data and the narrative fields (Figure 5.1), directly provide the text of the report (Figure 5.2)) or provide a direct reference (id) of the report.

The screenshot shows the *timePlot* web application interface. At the top, there's a navigation bar with the *timePlot* logo and user options: 'Connecté en tant que : Admin', 'Mon Compte', and 'Se déconnecter'. Below this is a search bar with the text 'Rechercher des événements par:' and three tabs: 'Texte libre', 'Mots clés', and 'Numéro de dossier'. The 'Mots clés' tab is active, showing a 'Query builder' section titled 'Entrer des critères de sélection d'un rapport ?'. This section contains various criteria with dropdown menus and input fields, such as 'Mot-clé titre', 'Mot-clé texte', 'Type A/C', 'Catégorie A/C', 'Immat. A/C', 'Catégorie d'Occurrence', 'Terrain de départ', 'Terrain d'arrivée', 'Exploitant', 'Type d'Exploitation', 'Aérodrome', 'Rapport partagé', 'Survenus après le:', and 'Survenus avant le:'. Below the query builder, it says '88 résultats'. There's a button 'Télécharger l'intégralité des résultats: Excel'. Below that is a table with 10 entries, showing columns: Id, Titre, Date, Type A/C, CatOcc, Aérodrome, and Exploitant. The table lists various aviation incidents, including 'NUAGE CENDRES VOLCANIQUES', 'RECALAGE HORAIRE INOPPORTUN SUITE NU...', 'DEROULEMENT FDF SUITE PRESENCE NUAGE...', 'ENTREE DANS UN NUAGE DE CENDRES VOLC...', 'PPV : MANQUE D'INFORMATION CONCERNAN...', 'SUSPICION DE NUAGE VOLCANIQUE', 'ROUTE SURVOLE NUAGE VOLCANIQUE', 'ENTREE DANS UN NUAGE DE CENDRES VOLC...', 'PASSAGE SUPPOSE DANS NUAGE D'ORIGINE...', and 'QUALITE INFOS SUJET PRESENCE NUAGES C...'. At the bottom, it says 'Showing 1 to 10 of 88 entries'. The footer of the application shows 'timePlot - 2013 - by Safety Data™ for dgac'.

Figure 5.1: *timePlot* GUI: source report selection *via* search

Figure 5.1 shows the search engine tab for selecting a source report. Visible in the upper half of the screen is the *query builder* that lists the criteria

available for choosing the source report. They correspond to keyword queries in the narrative fields (title and text) and to different³ metadata attributes. The criteria are joined by a logical AND. A document must satisfy all the criteria in order to be shown in the result list. The list itself is visible on the bottom half and is simply a table showing the title of the report, the date and several metadata attributes. In the example shown in Figure 5.1 the user is searching for documents containing “volcan”⁴ in their title.



Figure 5.2: *timePlot* GUI: source report selection *via* direct input

Figure 5.2 shows the input field where the user could paste the text of an existing report and find similar reports. This action essentially bypasses the selection interface (fig. 5.1) and shows the similarity page with the user text as source report. Initially this feature was intended to be used in order to circumvent the slow update cycle. The users wanted to be able to search for similar reports using data, which had not yet been imported into the system, as source reports. However, as we will see in the next section (§5.5) this feature was also used as a classical full text search engine, with several query terms rather than full report narratives.

After the source report has been chosen or entered by the user, the system identifies similar reports and presents them, alongside the source report, in the form of an interactive scatter plot (Figure 5.3). On top is the source report. Underneath is the scatter plot. Time is represented on the X-axis and similarity to the source report on the Y-axis. Each point on the plot represents a similar report.

³The attributes that can be used are chosen by the client.

⁴volcano

Hovering⁵ on a point in the scatter plot displays the corresponding report's title and underlines in yellow the words in common between it and the source report. This feature allows the user to quickly understand in what way the two reports are similar and was much appreciated by the users.

Clicking on the point opens a pop-up dialog with the report in question and the common terms underlined in yellow. (Figure 5.4). In the pop-up dialogue, a "Plot"⁶ button allows him to refocus the report in question as the source report effectively allowing the user to navigate between reports in an exploratory manner.



Figure 5.3: *timePlot* GUI: interactive scatter-plot

⁵Passing the mouse pointer without clicking

⁶We did not give much consideration to the naming of this button but it happened that for the users at the DGAC it became synonymous with the action of “displaying a document in the *timePlot* tool”. Thus a new French verb, “plotter” was created and is currently used by the people using the tool at that particular agency.

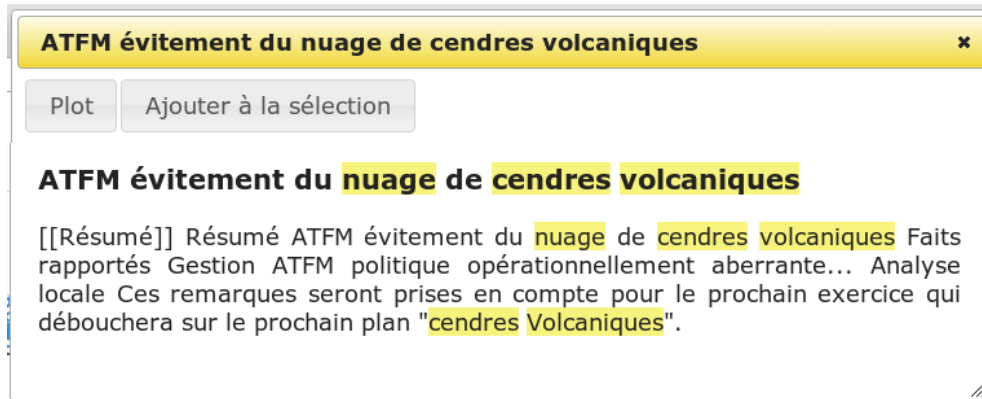


Figure 5.4: *timePlot* GUI: pop-up dialog of a similar report

Under the scatter plot, a *trend-line* (visible on figs. 5.6 and 5.5) represents the variation in frequency of the similar reports over time. Together with the overall distribution of the points on the plot, these two provide the user with information related to the behaviour of a given risk over time.

To compute the trend-line, we divide the overall temporal range in a fixed⁷ number of periods. The score corresponds to the sum of the similarity scores of the documents in each that period. The values are normalised to their *z*-scores and represented as a smooth line chart.

We will now see three examples of different chronological distributions of risky scenarios, showing the usefulness of the temporal dimension for organising the results.

5.3 Chronological distributions of risky scenarios

Here are three examples of recurrent incidents with different chronological distributions: A punctual incident, seasonal events and, most importantly an emerging risk.

5.3.1 A punctual incident

Figure 5.3 is an example of a punctual event in time. The source report concerns volcanic ash. The cluster of similar reports around May 2010, reflects

⁷The value is fixed manually anywhere between 30 and 100 depending on the range and the desired smoothness of the trend-line

events around the eruption of the Eyjafjallajökull volcano in Iceland, which caused disruptions in European air travel. Naturally, when the ashes dissipated, the problem disappeared. Although the event in this case is highly visible (both literally and figuratively) and well-known, quickly identifying such “peaks” in distributions is key to rapidly taking corrective actions.

5.3.2 Seasonal events

In figure 5.5, the source report concerns a *bird strike*, the aircraft collided with a bird on take-off. This is a very common type of occurrence. However, when we examine its temporal distribution, we can clearly identify a pattern. Most of the occurrences are concentrated in the warm months of the year (in Europe). This example of *seasonality* is not surprising as birds naturally tend to be less active in winter.

Figure 5.5: *timePlot* GUI: Example of seasonality - bird strikes

5.3.3 An emerging risk

Figure 5.6 is an example of an emergent risk. From roughly 2008, relatively cheap and extremely powerful laser pointers became available for purchase on the Internet and in some specialised stores. Probably driven by the undeniable awesomeness of these devices, people bought them only to be confronted by their also so flagrant lack of practical applications⁸. Frustrated by the aforementioned imbalance, some owners of such devices seem to routinely find solace in illuminating approaching aircraft from afar. This destructive be-

⁸The pointers are originally destined at amateur astronomers and used to make some pointy calculations on the distortions caused by the earth's atmosphere.

haviour has the potential to severely disrupt the delicate approach phase and even permanently blind the pilots. While placing it on the line between extreme stupidity and terrorism is up to the courts of law, it is a fact that since 2008 several thousand of occurrences are documented in France alone and barriers (such as anti-laser goggles for pilots) are being developed in the US.

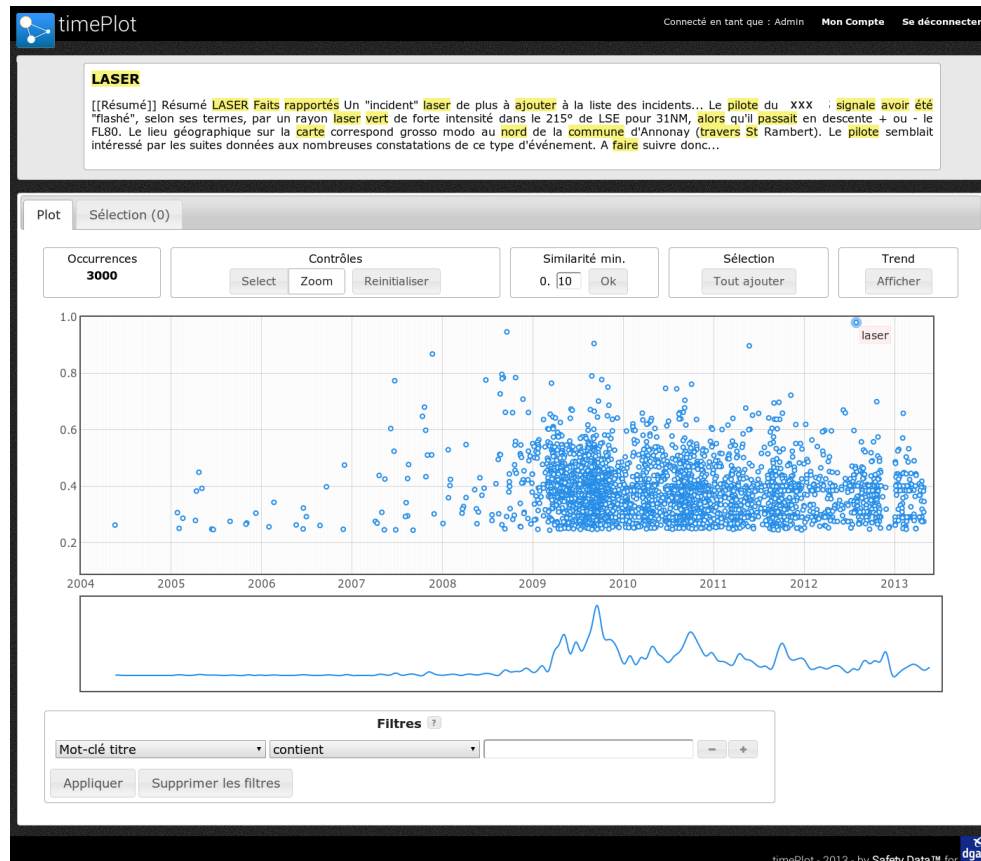


Figure 5.6: *timePlot* GUI: Example of emerging risk - laser pointers

Figure 5.6 also illustrates the built-in transparency of the system. When hovering on a point on the plot, the system dynamically highlights words that the reports share. Also, when a similar report is opened in a pop-up dialog (Figure 5.4), shared words are highlighted in both the source and the similar report. Besides providing an intuitive way for the user to determine if the report is of interest, this feature also provides information about the reasons

that a high similarity score was computed. Incorporating transparency in the system as a design choice partly echoes the high-recall requirements (3.1.3.3) expressed by the users. Giving information about the precise reasons a report is returned makes discarding noise much easier.

5.3.4 No chronological pattern

Figure 5.7 shows the tool with a single term, “souffle”⁹ used as a “source report”. This is a real query submitted by a user and illustrates the misuse of the tool we discuss in the next section. Also the results present no temporal pattern. We can see two vertical clusters. The one higher up corresponds to documents that have the term “souffle” in the title sections (as they are given more weight by the system). The points lower on the graph are documents that have the term only in the body of the document. The vertical pattern is an artifact of the rounding of the similarity score performed by *Lucene*.

Even if the tool was not intended for this kind of use, we can see how the scatter plot visualisation gives a much more concise view of the results and allows more documents to be returned. Currently the limit is set to 3000 documents. Thus much higher recall is possible. This is appreciated by the users who have shared with us that they prefer noisy results than ones with a lot of silence. To put it in other words, having to filter through false results is preferred to not finding true ones.

⁹Jet blast



Figure 5.7: *timePlot* GUI: Example of a query with no pattern

5.4 *timePlot* in use

TimePlot has been proposed to aviation safety experts at both the national (France) and European level. It is currently in active use in the French DGAC and in large national airline's safety intelligence service integrated in their safety management system. The tool handles respectively over 400,000 and tens of thousands of documents at these two clients.

At the DGAC, where the tool is at a most mature stage, there are currently 162 active users. Data is synchronised with their ECCAIRS database on a weekly basis and we have had a largely positive feedback from the users. The tool provides a much-needed workaround the inherent drawbacks of working with a imperfect ECCAIRS database. Where in a “perfect” ECCAIRS world the occurrences are thoroughly coded and thus the coded data is used as an entry point to the system, in the reality of the DGAC's database of nearly half a million occurrences, most of the valuable information still resides in the narrative parts. As ECCAIRS is, by design, not oriented toward using the

narratives, *timePlot* provides ways to quickly and easily find relevant information.

The DGAC are also starting an occurrence data sharing program supported by the tool. The service providers (airports and companies) willing to share part of their incident data will get free access to the tool with all the data that other operators participating in the programme have shared. Currently there are 167 active users and 560 monthly queries are performed at the DGAC.

One interesting scenario concerns the airline's testing of the tool. As part of the test we had provided the tool loaded with a database of publicly available incident reports. One of the questions that the safety officers were interested in concerned events that occurred at some of their diversion¹⁰ airports. For one particular airport in central Russia, the tool shed light on a larger than normal concentration of runway overruns - cases where the landing aircraft did not manage to stop in time. The problem was related to improper drainage of the runway surface and the company updated the procedures for landing there in case of emergency according to these findings.

In another case the experts were asked to investigate a series of specific incidents. The identification of similar incidents over an extended time period allowed them to determine that the original cluster was "a statistical accident" and not a developing trend, thus avoiding the (very costly) creation of a special investigative task force.

5.5 *timePlot* in misuse

The *timePlot* system was initially developed as a proof-of-concept but due to the positive reactions of *CFH - Safety Data*'s clients was quickly upgraded to a fully functional product. Being a prototype and proof-of-concept, a number of the design decisions were not really thought through. At the same time, however, the system was self-contained and provided an end-to-end solution from data-intake to a usable front-end GUI. This very quick "promotion" from a proof-of-concept to an industrial product provided us with a rather unique understanding of the users' needs, as the system started to be used in a number of unintended manners, way beyond the initial design specifications. Monitoring how the users "hijack" the system provided us with unique insights on the real world problems they were addressed using this system.

Given that there was no automatic update capabilities, new data was imported manually on a weekly basis. Anticipating the situation where a given report is considered interesting by a user, but is not yet present in

¹⁰ Airports to be used in case of an emergency. Having accurate and up to date information about these airports is problematic for companies, as they do not use them during normal operations. At the same time, the need for such information is of paramount importance when performing an emergency landing.

the system, we provided a “free text similarity” feature (fig. 5.2) where the user could copy a narrative from another source and paste it in the *timePlot* interface and search for similar incidents within the indexed data. We noticed two interesting and unintended use-patterns of this feature. The tool was used as a full-text search engine¹¹ and it was used to model a given accident scenario.

5.5.1 *timePlot* used as a full text search engine

Rather than pasting whole narratives, some users started using it more like a full-text search engine. The user would type in several search terms and then explore the results on the chronological scatter plot. This identified the need for such tools within the industry, where current solutions like ECCAIRS back metadata based exploration and undermine the textual information contained in the narratives.

A search query such as “approche non conforme ANC”¹² basically takes advantage of the indexing done for calculating similarity between documents and the dynamic highlighting of the input terms (fig. 5.6) and allows the user to quickly scan the collection for documents containing any or all of the terms. The fact that the user entered both the developed form and the acronym (ANC) clearly shows that the user is aware that the data is noisy and that the reports he is interested in may contain either of the variants.

A similar tactic was observed in the logs of the version deployed at the airline, where queries like “souffle jet blast” would be formulated to search in the narratives of both the reports in English and those in French. This lead us to search for the methods that allow cross lingual support, that we present in Section 6.3.

A use case scenario, related to regulation about the use of mobile phones on airplanes exemplifies this trend. The regulation had recently changed and led the company to consider allowing their use in the cockpit by the pilots. Using the tool, they searched for reports about possible interference (by essentially putting-in related keywords such as “interference”, “cockpit” and the names of different systems), and found one case where a mobile phone of a passenger seated in one of the front rows interfered with crucial instruments. Based on this, it was decided to maintain the ban in the company’s standard operating procedure.

¹¹While technically a misuse of the tool with regard to its initial purpose, we have to mention that this is actually the *intended* purpose of the *Lucene* search engine, we ourselves had “hijacked” to allow the tool to scale-up to the quantities of data it now processes.

¹²Unstabilised approach

5.5.2 Filtering similarity and modelling aspects of scenarios

We noticed that in some cases this functionality was not used to find a scenario, but rather to model a certain aspect of an incident. Users would input a variety of semantically related words or variants in the full-text field essentially using the system as a (crude) full-text search engine. After calculation, the users would scan the scatter plot, identify reports not matching their initial need and try to filter them out with keywords and Boolean operators using the system's filtering functions.

A user would, for example enter *fatigue*, *tired*, *rest*, and *sleep* in the full-text field. In this example the user tries to identify reports where fatigue was a factor. Afterwards, when looking at the results the user would notice that some reports mention "*metal fatigue*"¹³ and then apply a filter excluding the word *metal* from the results. The realisation that one of the initial terms (*fatigue*) is ambiguous, and that searching for it yields irrelevant results would come when looking at the results after the first iteration and not be expressed in the initial query. This type of narrowing down of the search criteria and progressive specification of the information need through query reformulation is typical for modern information seeking strategies (Jansen et al., 2009).

Another use case we noticed and found interesting was the following: Using the full text input area *as intended*, a user would paste the text of an incident report. He would then however start deleting terms from the text (and sometimes adding others) thus manually altering the input in order to "mask" a certain aspect of the incident. Based on this observation, we researched the possibility to automatically identify (and mask) "dimensions" of similarity based on correlation between terms and metadata attributes (§6.2).

This example shows how a clear understanding of both the tools and the data they manipulate allows the users to devise more intricate strategies to satisfy a given need for information. The *timePlot* tool, not being designed with such a use in mind naturally does not yield optimal results. However the fact that it was used in such a way clearly indicated that such needs must be addressed with a purpose-built system.

Such behaviour from the users is understandable in the sense that the information they seek is ever more elusive. The term "*non-technical signal*" came about in one of our discussions, making a distinction between *technical* matters that are clearly identifiable with simple terms (such as the names of specific components) and *non-technical* matters, such as human factors issues where key-word approaches are not powerful enough to reflect complex issues such as *confusion* or *distraction*.

In the end, whereas modern search engines put the emphasis on precise results with minimal engagement, we observed that in the context of searching for complex issues in noisy textual data, a human could not be expected to produce a coherent enough query *ex nihilo*. The tools, rather than simply

¹³*Fatigue* is used to denote the weakening of a material under forces.

aiming at the best possible result, should let the user “build a relationship” with the data they manipulate. After a first (underspecified) query, the tool ideally would give a picture of both signal and noise and allow the user to build on that.

This use made us consider ways in which to actively engage the users, and lead to the approach presented in section 6.5.

5.6 Lessons learned

Building and deploying the *timePlot* system provided us with a number of valuable lessons. In essence the application went from an early prototype to a production tool in very little time and with very little consideration for both technical issues, such as scaling with data, and usability. We basically provided the clients of *CFH -Safety Data* with several novel functionalities and let them make best use of the system. The fact that the tool was embraced shows how much the (rather basic) functionalities of the prototype were needed in the safety community. Way more important though is how, in the absence of proper prototyping and testing, the initial functionalities were adopted by the community and put to use in different ways than the intended ones.

The most important is how the unclear distinction between search in the classical information retrieval sense (§3.1) and similarity gave way to elaborate search strategies and a (partial) answer to a variety of information needs. We, as designers of the tool, sought to understand both *what* needed to be done to fully take these needs into account and *how* such a system should address the processing of the text. The next chapter elaborates on several more complex (*second order*) vector space representations that might provide part of the solution to more powerful tools in the future.

Another lesson we learned has to do with the usability of a given system. Somewhat intuitively at first, we wanted to make the tool as easy to use as possible and, conscious of the noisiness of the input material we sought to provide as much transparency as we can in the system. We discovered through the use that this type of functioning is preferred by the users than a more classical *black-box* approach. Rather than seeking perfect results and risking the inevitable cases where the system outputs something not right at all, our limited resources when building the system made us choose a solution where the tool embraces the noisiness of the data and the crudeness of the processing by seeking to communicate them to the user. This way they build a relationship with the system where, by understanding the underlying processing they in effect develop a much higher tolerance for the occasional “bad” or impertinent result.

But all in all the system was an industrial success. We conducted several interviews with end users and the verdict was unanimous: such a system benefits the experts working with occurrence data. This gave *CFH - Safety*

Data clues to the next generation of tools destined at the exploration of large databases of incident and accident reports.

5.7 Chapter conclusion

In this chapter we presented the *timePlot* system designed for identifying similar accident reports based on their narratives and grouping them based on a chronological criterion. The system was an industrial success. We observed the actual uses of the system and conducted interviews with its users and identified patterns of usage for which the system was not initially designed. Based on this we further refined our understanding of the needs of the aviation safety community and four aspects emerged: the need for a full text search engine, the need for multilingual support, the need for filtering different aspects (or dimensions) of the similarities the tool identifies and, finally, the willingness of the users to engage in a modelling of an accident based on examples.

In the next chapter we will show different approaches addressing these needs.

Chapter Six

Dimensions of similarity: from simple lexical overlap to interactive faceting and multilingual support

“Mad Hatter: “Why is a raven like a writing-desk?”

“Have you guessed the riddle yet?” the Hatter said, turning to Alice again.

“No, I give it up,” Alice replied: “What’s the answer¹?”

“I haven’t the slightest idea,” said the Hatter”

— Lewis Carroll *Alice’s Adventures in Wonderland*

This chapter explores the notion of similarity from several different angles. It is a collection of four independent approaches, each addressed at a different aspect of the complex notion. In Section 6.2 we present a method that learns from documents and their associated metadata attributes and allows to filter out one or another aspect of similarity. Next, in Section 6.3, we address the question of multilingual databases and explore the potential of second-order similarity methods to provide coherent representations of collections containing documents written in different languages. We compare the result of *Topic Modelling* to the information in ASRS’s metadata in Section 6.4. Finally, in Section 6.5 we present an approach based on *active learning*, allowing a user to model a certain aspect of an accidental scenario by providing the system with a few examples.

¹"Because it can produce a few notes, tho they are very flat; and it is *nevar* put with the wrong end in front!"

6.1 Chapter introduction

In the previous chapter we presented the *timePlot* system and how its initial designation was “hijacked” by the users to address several different informational needs. Independently we also used the system to explore the subject of document similarity. We were, on the one hand fully conscious that there is a lot more to modelling resemblance between two documents than can be captured by simple lexical overlap, such as the linguistic phenomena discussed in Section 3.1.2 and the faceted nature of similarity itself, which we will discuss in the next section. On the other hand, we also looked at how different processing methods might apply to the real world user needs, both in terms of expected output (multilingual similarity) and in terms of interacting with the system and the data as in the active learning approach presented in Section 6.5.

Let us also note that by naming this chapter “dimensions of similarity” we play on the fuzziness of the term “dimensions” that, for us, reflects the kind of intertwined considerations involved with hard questions we are trying to answer, namely how do we devise a tool that supports the pattern finding activity in which safety experts are involved when working with incident and accident data. In this sense “dimensions” denotes both the high dimensionality of the data in the strict sense of vector space models (§4.2) and, looked from the viewpoint of an expert, the fact that there exists not one similarity but an infinity of them. Thus, the pattern-finding exercise may be reformulated as finding just the right thread of similarity that weaves multiple occurrences into a coherent whole and provides a safety expert with the sort of “big picture understanding” that constantly improving safety depends on.

This chapter collates several independent takes on the subject of modelling similarity. The methods are, however related as they all build upon a simple term-document matrix or then apply some sort of transformation to produce a second vector-space representation of the reports. The methods listed here are mostly incompatible but they sweep broadly the extent of possibilities that we are offered when working with this data in this context.

The chapter is organised as follows: We start by discussing how representing document similarity by a single score confounds different aspects of their relatedness, which an expert might need to differentiate. We show that the major aspects of similarity are probably already captured by the metadata and in order to address this type of situations, in Section 6.2 we present a method that learns from documents and their associated metadata attributes and allows to filter out one or another aspect of similarity.

Next, in section 6.3, we address the question of multilingual databases and explore the potential of second-order similarity methods, to provide coherent representations of collections containing documents written in different languages. The method we use is however not limited to an interlingual context.

We also show how we can obtain much needed resources and inject knowledge into the system using readily available industry specific corpora.

In Section 6.4 we present how a currently popular dimensionality reduction technique (§4.3), *Topic Modelling*, has the potential to produce less noisy and more coherent representations of the major axes of variation within a given collection. We compare the produced *topics* with the rich metadata of the ASRS database and show that, while they overlap significantly the method also captures aspects of *topicality* not reflected by the metadata.

Finally, in Section 6.5 we present an approach directly building on the lessons learned from the *timePlot* system and after several years of refining the users needs. We use an active learning approach that allow users to model a certain aspect of an accidental scenario by providing the system with a few examples. A supervised learning algorithm then builds an initial model based on the examples and the users are encouraged to further refine it by validating and invalidating the relevance of the documents that the system identifies. We will present the initial specification and research around the method as well as a real-world scenario where the method was used to quantify crew fatigue and produce meaningful KPIs from a collection of incident reports.

6.2 Filtering aspects of similarity based on metadata

The similarity we use in the *timePlot* tool is based on lexical overlap. The more words in common, the more two documents are similar. This approach does not account for the fact that in many cases there exist (relatively) well coded metadata attributes capturing important information about the occurrence. Also, in our observation of the uses of the *timePlot* tool, we came by examples where users will delete terms from the text of a source report, deeming that this particular aspect of the scenario is not important. This lead us to consider a more global approach where we would exploit the informational overlap between metadata and text and use the former as a filter, in order to capture from the text only those aspects of similarity that are *not* presented in the metadata. This work was originally presented in (Tulechki and Tanguy, 2012).

6.2.1 Overlap between textual similarity and coded data

Let us start by illustrating the multifaceted character of first-order similarity with a constructed example.

Consider the following collection, comprising three short “documents”, each describing an incident.

1. “Bird-strike on takeoff”
2. “Turbulence on takeoff”

3. “Bird-strike on landing”

If we are interested in the first text and want to identify similar occurrences, the system will identify both documents 2 and 3 as similar, with a score of 0.5, as they both share one token with the first document.

From an expert’s point of view, however, these similarities are quite different. He would instantly differentiate between the pair 1 and 3 where a similar *event* occurred and 1 and 2 where completely different *events* occurred in similar *circumstances*. Similarity is, in a way *faceted*. While the expert might accept² such behaviour from the system, in a real world analysis scenario, it will be helpful to be able to filter out the facets.

Besides, these facets of similarity are already reflected in the coded data. In ICAO’s ADREP taxonomy (§2.2.3.5), for example, separate branches concern the *flight phase* and the *occurrence category*. The *flight phase* will indicate at which moment³ the event occurred. The *occurrence category* is a list of 36 values, classifying events on a macro level. It happens that both bird-strikes and turbulence encounters are sufficiently frequent as to have dedicated *occurrence categories*. The three documents in the example will then be coded as follows:

	Occurrence Cat.	Flight Phase
Doc 1	BIRD	Takeoff
Doc 2	TURB	Landing
Doc 3	BIRD	Landing

In order to measure this overlap more precisely, using information about occurrence category and flight phase, we constructed a test corpus of 482 documents about turbulence encounters and bird strikes. As both events can occur on landing and on takeoff, we balanced the corpus as to have an even distribution of documents on both flight phase and occurrence category:

	TURB	BIRD	Total
Landing	118	133	251
Takeoff	107	124	231
Total	225	257	482

We then computed a standard⁴ term-space similarity matrix for the set and we calculated an average overlap score by taking for each document the 30 most similar documents and comparing their metadata with that of the source document. The results show a significant overlap: Almost nine out

²Given that transparency is incorporated in the results so the user understand the reason a given result is produced.

³from the parked position before the flight, through cruising at 35000 feet to the parked position after the flight

⁴We used the processing chain from the *timePlot* system prototype (§5.1)

of ten (89%) similar documents share the same occurrence category and 75% share the same flight phase. Given the balanced nature of the corpus, if no overlap was present, we would have expected these numbers to be around 50% in both cases.

6.2.2 Smoothing the major facets

Our next goal was to isolate the major facets of variations and smooth them out. Put in other words, we do not want textual similarity to convey information that is already present in the coded data and, in a way, emphasize the complementarity of the two.

The first step is to link the descriptors used by textual similarity - the terms and the coded data, using an interdependence measure, PMI⁵ (see (Manning et al., 2008, Section 13.5.1) for the exact algorithm used). In IR, given that some form of human categorisation of the collection exists, such *feature selection* methods, are used to reduce the term-space, keeping only those terms that are statistically correlated with a given class.

The underlying hypothesis in IR assumes that any human classification of a document collection is a valid source of information about the meaningful variation within the specific domain and therefore the terms highly associated with classes will most likely be less noisy and more valid descriptors when used for indexing the same collection.

In our case, we want the exact opposite. Looking to artificially *decorrelate* textual similarity from the coded data, we will subtract the highly associated terms from the term-space.

First, using 4450⁶ documents we calculate PMI between each term and each class. The five most correlated terms are presented in table 6.1.

	Occurrence Cat.		Flight Phase	
	TURB	BIRD	Landing	Takeoff
1	vent	aviaire	approche	décollage
2	turbulence	collision	finale	poussée
3	gaz	oiseau	atterrissage	rotation
4	arrière	impact	stabilisation	t/o
5	windshear	bird	arrondir	vr

Table 6.1: Highly associated terms for each class

Using this information, we can now selectively filter-out terms associated with either the flight phase (FPh) and/or the occurrence category (OccCat),

⁵Pointwise Mutual Information

⁶All the documents coded with one of the 4 classes in the DGAC's collection.

1002. FILTERING ASPECTS OF SIMILARITY BASED ON METADATA

based on a threshold. And calculate and compare the similarities in the filtered and unfiltered matrices.

Table 6.2 shows average overlap (AO) for both unfiltered similarity and filtered similarity. We also calculated a mean disturbance rate (DR) representing, on average, the number of new documents in the top 30 similar documents when a filter is applied.

	AO FIPh	AO OccCat	DR
Unfiltered	75%	89%	-
Filtered for FIPh	64%	84%	9,8
Filtered for OccCat	73%	69%	13,6

Table 6.2: Filtered and unfiltered mean overlap between textual similarity and coded data

We can see that applying a filter for a given facet reduces the number of similar documents which share that facet with the source document. For an incident report concerning bird-strikes on take-off, at average 89% of the 30 most similar reports will be about bird-strikes and 75% of the top 30 , will concern events occurred at take-off. When we filter the occurrence category, the average number of similar reports concerning birdstrikes drops to 69%. At average there are 13,6 new documents in the top 30.

Let us look more in detail at what this disturbance contributes from a qualitative perspective and how such a system has the potential to identify minor secondary facets of similarity. In our test corpus we identified the following document:

INCURSION VFE SUITE CISAILLEMENT EN FINALE.

Fort cisaillement en finale reporté par les avions précédents. La soudaineté du phénomène surprend l'OPL PF. Légère incursion dans la VFE (3 ou 4 kts). Réponse des commandes par CDB (double pilotage pendant 1 à 2 s.). Avion stabilisé, l'OPL reprend les commandes. Atterrissage sans problème.

It is an occurrence concerning turbulence encounter on short final. However it mentions also a *double input*⁷ event.

When looking for similar documents without any filter it comes to no surprise that the most similar reports concern turbulence encounters while landing. This is the most-similar document⁸:

⁷ *double input* is an means that both pilots acted on the controls simultaneously.

⁸Shared terms are underlined

FORT CISAILLEMENT DE VENT EN FINALE 26R CDG.

FORT CISAILLEMENT DE VENT EN FINALE.

However when filters on **both** *flight phase* and *occurrence category* are applied, documents that share only these facets of similarity will naturally appear further down the list of similar documents and shared secondary facets, such as the “double input” event, will be emphasised and contribute more to the similarity score.

BREF DOUBLE PILOTAGE AU DECOLLAGE.

OPL PF au décollage. Vent travers avec rafales. Brève action réflexe en latéral du CDB pour contrer rafale et début d’inclinaison à droite. Prise de priorité peu pertinente pour effet immédiat.

This technique shows how we can exploit the taxonomy of a collection by linking the textual features to specific metadata categories and then using these links to influence the behaviour of the similarity computation. In order to extract the links we need a reasonably large collection of coded reports, but once the links are computed, the method is also applicable to uncoded documents and separate similar collections.

6.3 Computing interlingual similarity

In this section we will present a *second order similarity* technique for calculating relatedness between documents written in different languages. This work was originally published in (Tulechki and Tanguy, 2013).

A significant hurdle to adequate processing of incident reports stems directly from the intrinsically international character of aviation itself. Information about incidents comes from variety of sources and, even if English is the operational *lingua franca* of choice, incidents are still often reported in other languages without translation. While this is an evident problem when aggregating data at higher levels, it can even be an issue at the level of a company’s own internal reporting. Such is, for example, the case with national airlines’ internal reporting systems, where the bulk of the data is reported in the nation’s official language, but as some of the personnel are not native speakers, English is also permitted. The result is a multilingual incident report database where both languages coexist. This is a major challenge both for simple applications such as search engines, and more complex applications such as identification of similar incidents.

As we saw in Section 4.3, a family of second-order representations such as ESA⁹ (Gabrilovich and Markovitch, 2007) or, more generally speaking, *vectorisation* (Claveau, 2012) can outperform simple term-space representations and present several key advantages related to concurrent second-order representation methods:

- **Robustness:** They do not require lexical resources specifically tailored to the specific document collection being analysed.
- **Interpretability:** Each dimension of the final vector-space represents the relatedness of a document with another one from the external collection. The second order representations is “explicit” (hence the *E* in ESA) and a human is capable of interpreting one by one the individual dimensions.
- **Control of the representation:** By relying on an external resource (document collection) to shape the final vector-space, this family of methods allows us to be free of frequency-based biases, where under- or over-representation of a certain class of documents in the collection influence the dimensions of the resulting second-order vector-space representation. Furthermore, by making the most of the interpretable character of the representation, we can influence the results by selecting both which documents from the external collection are used and how they are grouped together (Gabrilovich and Markovitch, 2007)

Another key advantage of the *vectorisation* family of methods is that, by relying on relatedness to documents rather than terms (or collections of terms), the final vector-space is, in a way, language independent. A coordinate on a dimension in this space is no more than the similarity between (the meaning) of two documents, and by assuming that the meaning of a text and its translation in another language are identical, we can effectively map documents written in different languages to the same space. We explored just this property of vectorisation.

In the original ESA-model, a collection of documents is mapped to a second order space where each document is represented by a vector of similarities with a set of *pivot* documents.

In the multilingual variant of ESA, called CL-ESA¹⁰ (Sorg and Cimiano, 2012), the pivots consist of sets of translations of the same document.

Figure 6.1 shows the general principle of CL-ESA. As in ESA, documents are represented as vectors of similarities with a collection of *pivots*. In the multilingual variant each document is compared to the subset of pivots written in the same language. Assuming that a document and its translation in another language are *semantically* identical, we can construct language-independent second-order representations.

⁹Explicit Semantic Analysis

¹⁰Cross Lingual Explicit Semantic Analysis

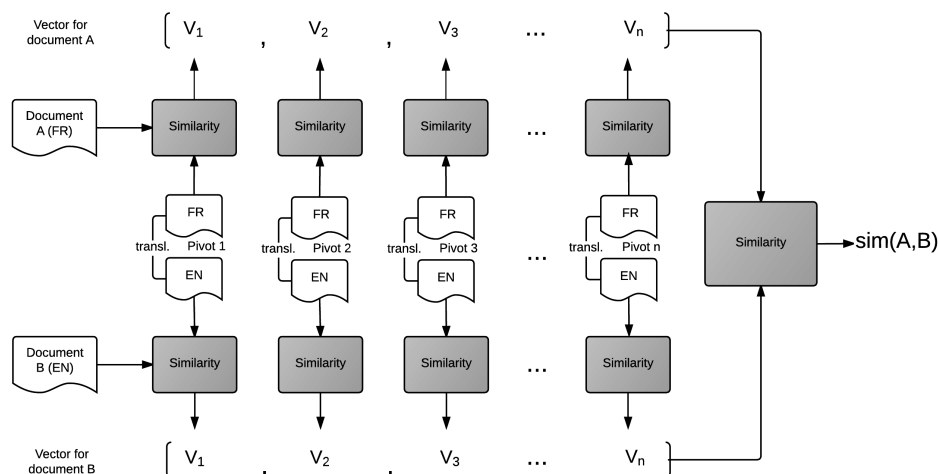


Figure 6.1: Cross-lingual ESA

6.3.1 Constructing the pivots

As Gabrilovich and Markovitch (2007) put forward, this family of methods allows computing a more *semantic* relatedness between documents, by mapping them to “natural concepts” rather than to noisy and ambiguity-ridden term-vectors. In the original implementations of both ESA and the cross-language variant (Sorg and Cimiano, 2012), the authors used articles from Wikipedia to construct the pivots. As a resource Wikipedia has a very broad thematic coverage and is well suited for general texts, like the press articles that Gabrilovich and Markovitch (2007) used to evaluate their system.

A resource, such as Wikipedia is however ill-suited to the highly-technical and narrow themed writing of documents treating of aviation incidents. Wikipedia’s concepts are too general to capture the fine topical variation presented in such documents, while at the same time being too-varied and thus noise-inducing.

An adequate set of pivot documents should, ideally capture most of the inherent variation in aviation incidents. While the online encyclopedia has articles on Aviation Safety and, some of the systems and technical components mentioned in incident reports, it does not cover concepts such as, for example, different alerts one sees frequently mentioned in the report narratives. At the same time, Wikipedia’s broad coverage entails that a vast amount of concepts are *a priori* irrelevant to aviation safety. Representing an incident report by its similarity (or dissimilarity) with the article about, say, Walt Disney is not informative at all.

Following the original ESA philosophy, an ideal pivot corpus must have documents representing all the relevant concepts potentially present in the

indexed collection. In our case such a resource simply does not exist. Fortunately, as Claveau (2012) demonstrates, second order similarity can achieve better results than traditional first order methods even when the corpus of pivots is a randomly assembled collection of texts, without any structure. To our knowledge, the question on how to construct the pivot corpus, in the context of a collection of domain-specific documents has not been explored.

With respect to the above mentioned considerations and heavily constrained by practical issues such as availability, we chose a solution half way between the (presumed) corpus of concepts of the original ESA implementation and an unstructured collection of texts.

We constructed the pivot corpus using official accident reports (§2.1.2.2) issued by the investigation authority of Canada, the TSB. Given that, in Canada, both French and English are official languages, accident reports are systematically published in both. They are generally long documents and have identifiable parts, each having a different discursive function. As a reminder, the beginning will usually consist of a narrative of the event. Later on, the analytical parts will “zoom in” and provide descriptions of the exact mode of failure of a given (human or mechanical) subsystem. It follows that, taken as a whole, accident reports can not be considered as representative of *concepts*. However, when broken up into smaller sections, each section represents a rather concise theme. The following paragraph from such a report, for example, has high internal coherence, explaining a particular aspect of the behaviour of helicopters:

Pushing the cyclic forward following a pull-up or rapid climb, or even from level flight, produces a low-G (weightless) flight condition. If the helicopter is still pitching forward when the pilot applies aft cyclic to reload the rotor, the rotor disc may tilt aft relative to the fuselage before it is reloaded. The main rotor torque reaction will then combine with tail rotor thrust to produce a powerful right rolling moment on the fuselage. With no lift from the rotor, there is no lateral control to stop the rapid right roll and mast bumping can occur. Severe in-flight mast bumping usually results in main rotor shaft separation and/or rotor blade contact with the fuselage.

It follows that, documents about out-of-control helicopters will have a high loading on the dimension represented by this pivot.

For this reason rather than using whole documents as our pivots, we segmented them into paragraphs (using formatting cues such as the `</br>` *html* tag) and then aligned them with their corresponding translations using the isomorphy of the *html* structure on the relative pages on the TSB website, thus obtaining a total of 10032 pairs of pivots from 390 accident reports.

6.3.2 Indexing the documents

Once the pivot corpus is constructed, processing and indexing documents is straightforward. As shown in Figure 6.1, each document is represented by a vector of similarities with the part of the pivot written in the same language. We first apply, the following processing chain to each document:

- **Language detection:** The language of the document is detected in order to determine which part of the pivots it will be matched against and the parameters of the following processing steps.
- **Tokenization:** The documents are *tokenized* (§4.1.1) using the tokenizer provided with Apache Lucene.
- **Stemming:** The tokens are *stemmed* using the *Snowball* stemmer.
- **Stoplist:** A standard *stoplist* is applied to exclude common tokens, such as determiners and prepositions.

The same chain is applied once to each of the pivots for them to be compatible with the indexed documents.

A first-order vector space representation is constructed and *weighing* (§4.2.2) is applied using the PPMI¹¹ method (Turney and Pantel, 2010). Then a similarity score is calculated between each of the documents and each of the pivots of the corresponding language in order to construct the second-order vectors.

We then calculate a *cosine similarity*, using the same method as in the original *timePlot* implementation (§5.1).

6.3.3 Evaluation in a multilingual context

In order to evaluate the system we devised a protocol similar to the one used by Sorg and Cimiano (2012). Using a corpus of preliminary reports and briefs (§2.1.3) from the CADORS database we constructed a test-set of 8394 bilingual documents. Each of these documents is a relatively short report published in Quebec. Like for the TSB accident reports, local regulation requires that the documents be translated in English and in French. Table 6.3 shows one such report.

Like Sorg and Cimiano (2012), we used the task of *mate retrieval*, consisting of searching for the translation (the *mate*) of a document within the *n*-most similar documents retrieved by the system. We first split each of the documents in the test corpus into two monolingual documents, giving a total of 16788 entries. We then construct a document-document (second-order) similarity matrix and, for each document, we calculate the *rank* of its translation in the list of the most similar documents returned by the system. If

¹¹Positive Pointwise Mutual Information

CRQ590M, a Beech A100 operated by Air Creebec as flight number CRQ590, was on an IFR MEDEVAC flight from Chibougamau/Chapais (CYMT) to Montréal/Trudeau (CYUL). At 1535Z, the crew was instructed to conduct a missed approach for Runway 06R due to the presence of C-FFWJ, an Airbus A-320 operated by Air Canada as flight number ACA407, which was lined up for departure and which had a mechanical problem. CRQ590 eventually landed without incident at 1546Z.	CRQ590M, un Beech A100 exploité par Air Creebec sous l'indicatif de vol CRQ590, effectuait un vol d'évacuation médicale selon les règles de vol aux instruments (IFR) depuis Chibougamau / Chapais (CYMT) à destination de Montréal/Trudeau (CYUL). À 1535Z, l'équipage a reçu l'instruction d'interrompre son approche pour la piste 06 droite en raison de la présence de C-FFWJ, un Airbus A-320 exploité par Air Canada sous l'indicatif de vol ACA407 qui était aligné au départ et qui avait un problème mécanique. CRQ590 a finalement atterri sans encombre à 1546Z.
---	--

Table 6.3: An incident report from the CADORS database

it is the first (most similar document) it will be at rang 1, if it is the 7th most-similar document, it will be at rang 7.

We used the metric of *recall at rang k* ($R@k$) to evaluate the overall performance of the system. $R@k$ represents the proportion of translations present within the k most-similar documents for the whole corpus. An $R@1$ of 1, means that for every document, its translation is the most similar document - the system is perfect. A $R@10$ of 1 means that all documents have their translations within the 10 most similar documents.

Table 6.4 shows the results of the evaluation task.

	FR	EN
$R@1$	0,43	0,45
$R@10$	0,71	0,74
$R@100$	0,90	0,94

Table 6.4: Mate retrieval results

As we can see, the results are encouraging. In more than 40% of the cases, the translated document was the most similar document returned by the system and in more than 70% of the cases it was within the 10 most similar documents. For comparison, Sorg and Cimiano (2012) report a $R@10$ between 0.27 and 0.52.

6.3.4 Discussion

This experiment demonstrated that the ESA family of methods is applicable in the context of multilingual databases of incident reports.

The experiments showed however several areas where further research has the potential to improve the results. The availability of an adequate multilingual resource for the pivots is crucial. While we achieved acceptable results by just taking paragraphs from accident reports as pivots, our intuition tells us that the *explicit* character of the method merits further investigation. Gabrilovich and Markovitch (2007) emphasises on the importance of having the “right” concepts and, accordingly, part of the work goes into investigating the “right” way to concatenate Wikipedia articles based on different levels of grouping in the categorisation hierarchy of the online encyclopedia. In our case we can ask ourselves how to smooth the pivot corpus in order to get a more “natural” set of pivot documents. This can be done in multiple ways.

One will be to provide more advanced methods of “cutting-up” the documents in conceptually coherent parts. The work we did with Campello Rodrigues (2013), aimed at *zoning* accident reports into sections with different rhetorical functions, provides an interesting starting point. Being capable of isolating relevant parts of these documents based on their overall rhetorical structure, only those zones, having well defined and context-independent informational content (such as the purely descriptive parts) could provide a less noisy corpus of pivots.

The *explicit* character of the method also has the advantage of being interpretable. Given that we can identify which of the pivots are contributing to a given similarity score, one can then extrapolate (for example by using standard document classification techniques (§3.2) and values from the coded data) which aspects of an incident are captured by a given cluster of similar documents, effectively addressing the same concerns discussed in the previous section.

The aforementioned considerations are valid for both intralingual and interlingual ESA-like methods. For the interlingual part only, the question of the availability of aligned resources is a central one. While we were “lucky” that the English-French pair is represented by the Canadian documents, aligned technical documents for other language pairs are not easy to come by. Currently one of the needs expressed by the aviation safety community in Europe (and carried by the European Commission) is putting order in the centralised incident repositories, where incidents are reported in all of the European languages. For a ESA-like method to be applicable, all the pivots need to be translations of the same (domain specific) texts in all official languages in the EU.

6.4 Topic Modelling applied to the ASRS database

In this section we will present an experiment with *Topic Modelling* applied to the ASRS (§2.1.4) database. This work is a summary of the Master’s thesis of Nicolas Ribeiro (2014) and was presented in (Tanguy et al., 2015).

6.4.1 Topic Modelling in a nutshell

Probabilistic topic modelling is a generic method initially designed by David Blei (Blei et al., 2003; Blei, 2012). Following older methods of documents representation such as Latent Semantic Indexing (Deerwester et al., 1990), its main purpose is to represent a collection of documents in a vector space with a reduced number of dimensions or *topics* (as opposed to traditional vector spaces where each dimension corresponds to a single term or word). These topics or latent dimensions are calculated without any kind of supervision or external knowledge, based solely on the distribution of words in the documents. Thus, the topics are supposed to be a good representation to the underlying thematic structure of the collection.

The statistical techniques behind topic modelling make a number of assumption that can be summarised as follows: a document is essentially a set (or bag) of words; a document expresses a number of topics of varying importance according to a specific distribution; a topic is expressed with words according to a specific distribution. Thus, by observing a collection of documents, one can empirically estimate the two distributions (document-topic and topic-words) that fit the observed frequencies of words in documents. The basic version of topic modelling details this crudely defined method by selecting a well suited distribution (Dirichlet, hence “Latent Dirichlet Attribution” the name of the most widely used version of topic modelling) as well as the algorithms that can estimate the actual parameters.

From a practical point of view, given a collection of documents (essentially their decomposition as bags of words), a fixed number T of topics and a few hyper-parameters, a topic modelling session produces two matrices.

The first one is a document-topic matrix in which each document is described as a vector across the T topics. In other words, it tells us what topics are the most important ones for each document. This information can be used as such for indexing and comparing document within a smaller vector space.

The second matrix is a topic-word matrix in which each of the T topics is represented as weights associated to each word. In other terms, it gives the words most frequently associated to each topic. This information can be used to interpret the topics and enable a user to get a readable description of a document in terms of topics. The following experiments details are as follows, although they are presented more thoroughly in Ribeiro (2014). We used a collection of 167,350 documents from the ASRS database (from 1987 to 2012), and extracted the narrative parts for a total of 17 million words. We used the TreeTagger part-of-speech tagger to use word lemmas instead of wordforms and to remove function words (prepositions, determiners, numbers, etc.). In order to deal with the language variation in the history of ASRS (as described in section 2.1.4), all technical words were replaced with their standard acronym (ACFT, WX, etc.). Finally, all tokens were folded to lowercase.

Topic models were computed using the Gensim library (Řehůřek and Sojka,

2010) using the standard method¹² (Gibbs sampling) with a target number of topics $T = 50$. Calculation takes about 2 hours on a 4-core 3.1GHz processor computer.

Although this method is non-deterministic, we could observe through several runs that the results are quite stable, as it has already been observed for corpora this size. The choice of 50 topics is arbitrary, but was finally chosen as the number for which interpretation of the resulting topics was the most satisfactory: we will now come to this crucial phase.

6.4.2 Interpreting the topics

As explained before, a topic model for a given corpus consists in two matrices, document \times topic and topic \times words. The “Main terms” column of information shown in table 6.5 comes from the topic \times words matrix. This column contains, for 5 sample topics¹³, the 15 words that have the highest probability of expressing it according to the Dirichlet distribution estimated from the observed word distribution. This information is traditionally used for describing a topic to a user and used for testing the relevance and cohesion of this representation (Chang et al., 2009).

#	Main terms	Expert	Metadata (R)
1	<i>rwby, twwy, taxi, hold, short, gnd, twr, clr, acft, tkof, line, clnrc, ctl, cross, pos</i>	Ground	anomaly:ground incursion (0.65); phase:taxi (0.65)
2	<i>day, hr, time, trip, crew, duty, flt, night, fatigue, rest, leg, fly, min, morning, late</i>	Fatigue	anomaly:company policy (0.11)
3	<i>pax, flt, attendant, cabin, smoke, capt, cockpit, seat, back, crew, acft, emer, told, smell, lndg</i>	Cabin	anomaly:flight deck/cabin (0.60)
4	<i>wx, ice, turb, flt, tstm, moderate, rain, icing, acft, severe, radar, area, light, encounter, condition</i>	Weather	primary problem:weather (0.45); anomaly:inflight event (0.37), component:weather radar (0.12)
5	<i>acft, checklist, flt, call, capt, maint, lndg, make, l, fo, flap, time, control, return, continue</i>	???	primary problem:aircraft (0.24); anomaly:equipment (0.24); detector:flight attendant (0.23); component:turbine(0.13); component:flap control (0.13)... (6 more)

Table 6.5: The 5 first topics extracted from the ASRS corpus

¹²The hyper-parameters were left to their default value: $\alpha = 1/T$, $\beta = 1/T$, 50 passes.

¹³The topics’ order is insignificant as it is an artefact of the randomisation process at the beginning of the modelling process.

A safety expert was presented the 15 most contributing words for each of the 50 topics, and was asked to describe in a few words what each of these topics could mean. His feedback is presented in the “Expert” column of table 6.5. For 43 topics out of 50 the expert was able to identify a theme or a small set of themes that could be expressed by the words with the highest probability values. Although some of the words may seem opaque to a layman, most of them are in fact quite transparent. Contributing words for topic 4, for example comprise both the overall category (*WX* is the standard acronym for *weather*), various meteorological phenomena (*ice/icing*, *rain*, *thunderstorm* (*TSTM*)), common modifiers (*light*, *moderate*, *severe*) or consequences (*turbulences* (*TURB*)); all this make it an easily interpretable topic. This is not the case for topic #5, where no coherence could be found, as the most contributing words are scattered across several aspects of flying an airplane.

The document×topic matrix provides another means for interpreting the topics: each document is represented by a vector of weights across the 50 topics. That means that each topic can be viewed as a distribution over the documents, and as such can be compared to the documents’ metadata. We thus computed Pearson’s correlation coefficient between each topic and each metadata value across the documents (considering 1 if the document’s metadata contain this value, and 0 otherwise). This gave us a different, more objective angle to interpret each topic, as we could identify which metadata value was the most strongly associated to each topic. These values are indicated in the “Metadata” column of table 6.5, along with the correlation coefficient’s score¹⁴.

First, we can see that for some topics (number 1, 3 and 4 in our selection) one or two highly correlated values (> 0.4) can be identified, and that these confirm the expert’s interpretation. Other attributes can appear as secondary correlates, such as flight phase and reporting person, but nevertheless it appears that such topics have captured a well-known aspect of incident reports. This is the case for 38 of the 50 topics. It has to be noted that any aspect of a report can be thus “captured” by a topic. For example, one particular topic was associated to flights in California, the contributing words being the names of locations in this traffic-dense area.

A second case is that of the topics that could easily be identified by the expert but do not show any marked correlation with the metadata. This is the case for topic 2 in our selection, where the only correlated attribute is the company policy, although with a very low score. This kind of topic is extremely interesting, as it shows that corpus analysis by this kind of method can make some aspects of incident reports emerge. Only 2 of these could be identified in the 50 topics examined in our experiment: *fatigue* and *flight*

¹⁴Only the attributes with a positive correlation higher than 0.1 are presented. This threshold was chosen arbitrarily as the population is too large to have non-significative correlations scores.

planning. It is important to note that the *fatigue* attribute was added to the ASRS taxonomy, along with other human factors, in 2009. Even though the subset it covers is too small for meaningful results, and is heavily biased because of this temporal constraint, partial analysis indicates that this topic is highly correlated to this attribute.

The 10 remaining topics could not be associated to any single aspect of reports. This is the case for topic 5 in our selection, where the correlated attributes are numerous and scattered, making no more sense to the expert than the contributing words. Other configurations in this category are topics for which several identifiable topics are mixed together, and which are split when a larger number of topics T is extracted.

6.4.3 Discussion

Although we only performed a limited number of experiments with topic modelling on incident reports, it appears that topic modelling is suitable for occurrence data. It is a very robust method that takes clear advantage of large collection of redundant documents as it is the case for incident reports. Most of the topics identified are in fact relevant aspects of these documents, as can be seen through an expert's interpretation. However, only a small fraction of identified topics are both relevant and independent from the metadata attributes, and as such provide an added value.

One of the main limitations of this approach is the granularity of the extracted topics, especially when it is compared to the level of details attained in the organised description and indexing of aviation incident reports. As seen in the previous analysis of the resulting topics, most of the topics do little less than confirm an organisation that is clearly expressed by some of the metadata. If in some cases this method can identify non-encoded aspects, they are difficult to detect among other unavoidably noisy topics. However, this technique can be extremely valuable for reports database that are not supported by a thorough classification scheme and extensive metadata. This can be the case of databases that need to be consolidated, or even for the replacement of an unsuitable taxonomy.

On the technical level, topic models are somewhat sensible to a number of parameters, the first of which is the requested number of topics. We performed several tests on the same data with $T = 10$, $T = 100$ and $T = 200$. None of the topics among the 10 were interpretable, as they all mingle several aspects of the reports. Interesting things happened with 100 topics, including the clear and expected separation of topics (from the 50 described above) that could be identified as an agglomeration of quite distinct sub-topics by the expert. However, this led to only a few such improvements, most other topics were deemed unnecessarily split. With the highest tested value (200), many resulting topics were related to geography, with high-weighted tokens corresponding to airports, beacon codes and city names (mostly in the US). Although these

topics were coherent and easily interpreted, their informational value seems quite low. Finally, we could identify a few very stable topics across the variation on T ; this is the case for topic 2 (related to fatigue) that was found almost identical in all experiments with $T \geq 50$. In the end, the optimal value for T cannot be evaluated without a complete and thorough interpretation of resulting topics, and is estimated to be highly dependent on the collection of documents.

Nevertheless, we see potential applications of the technique in both calculating similarity and as an initial step when designing taxonomies for large collections of textual reports, when a taxonomy is not available¹⁵.

6.4.4 Application to similarity

In order to apply topic modelling to similarity, one just needs to compute the similarity scores between documents based on their score for each topic. Such an application will essentially address the same issue we described in section 6.2, where the user will be able to “switch on/off” different topics and thus influence the similarities identified by the system. To build on the previous examples (table 6.5) if for some reason an expert is not interested in the role weather played in an incident he would turn off topic 4. Thus documents similar because of weather related issues will not be identified. Such a shift in similarity will hopefully bring to light another more subtle or hidden pattern in the data.

6.4.5 Application to uncoded collections

In determining the significant overlap between the produced topics and the metadata we can now consider topic modelling as a viable solution for treating large uncoded (or ineffectively) coded collections. Essentially the same interpretation exercise we performed in this experiment can be considered as a base for a first version of a coding taxonomy. An expert would be presented with the major topics and asked to interpret them. Thus an initial set of metadata categories can be established. Such an approach would also have the added value that, by definition, the categories will be easy to classify by automatic classification techniques as they will be reflected in the narrative parts.

¹⁵As is sometimes the case in industries and sectors where incident reporting is a more recent enterprise.

6.5 Active learning for interactive model construction

In this section we describe the intended approach, the algorithm we designed and a simulation we are currently running as part of *Safety Data's* R&D program in order to better understand the behaviour of the active learning approach and tune it before we submit it to real users. This work was done in collaboration with Assaf Urieli and is presented in (Tanguy et al., 2015).

We described in the previous section our observations on how the *timePlot* tool was used to model a facet of an incident (§5.5). This usage scenario and the successful performance of the machine learning approach described in section 3.2 led us to design a system that relies on the availability of an expert and use a variant of machine learning techniques: active learning (Olsson, 2009). These variants are based on traditional supervised learning methods, but take into account the fact that training data are expensive to get when an expert is required for labelling items. Active learning strategies try to make a smart usage of the expert's time by submitting to his judgement only the difficult or borderline items. This can only be done through an iterative process with a dose of interaction with the user.

6.5.1 An interactive approach to signal detection

The basic idea behind the system is to allow the users to model a given aspect of an incident by providing examples of documents that are related to the particular aspect. We start with the assumption that the aspect is partially identifiable by a query using a full text search engine and available metadata. A user interested in confused flight crews will presumably start by querying the system for documents containing the word “confusion”. This set will, however contain some documents that do not match the user's information need¹⁶. When looking through the documents, the user will notice this and would like to exclude them from the search. At the same time there will be documents that do not contain the word “confusion” and that are relevant. The system should also be able to identify such documents.

We have systematised this process into what we have call “creating a *Dimension*”. A *Dimension*, from a user's point of view is a dynamically created label that can potentially apply to any report in the corpus, as well as any new report introduced. Conceptually, creating a *Dimension* can be compared to introducing a new metadata attribute / value pair to an existing taxonomy. However, the difference is that we seek to render the process the least

¹⁶Consider documents speaking for confusing call signs, for example. AF259 and AF299 flying at the same time in the same area makes it rather hard to communicate with ATC over the radio but does not necessarily amount to the flight crews being confused about what they are supposed to do.

time-consuming as possible and not require extensive coding of all the existing reports.

From a system's point of view a *Dimension* is no more than a classifier that produces a yes/no partitioning of the corpus. The algorithm described in the next section shows the process for creating and training this classifier.

6.5.2 Active learning algorithm

The outline of our system is the following: we start with a rough estimation of what the expert considers as the target (positive) reports. We train a classifier based on this data, and then apply it to the entire collection. Due to the nature of classification algorithms (and their need for generalisation), this classifier provides a different set of positive reports. Using the error margin (or probabilistic confidence score) provided by the classifier, we can identify borderline reports, on both side of the decision: we select these few fairly positive and fairly negative items and submit them to the expert's judgement. Based on his decisions, we obtain a new approximation of his needs, and can build another classifier, and so on until the expert reaches a satisfactory result. This active learning principle is also called uncertainty-based sampling and has been proposed in a number of NLP tasks (e.g. Kristjansson et al. (2004) for information extraction, Roth and Small (2006) for semantic role labelling).

Algorithm 6.5.1 shows in details the active learning algorithm for training a Dimension. Given a corpus \mathcal{C} of safety reports, we wish to calculate a dimension vector \mathcal{D} assigning a dimension yes/no value to each document in the corpus. The expert kicks the system off by providing an initial approximate set of positive examples \mathcal{P} . These are either the result of a keyword search for keywords highly suggestive of the target dimension, a set of similar reports identified with *timePlot* or a handful of manually selected documents. The system also requires a set of training parameters which depend on the classifier used (e.g. C and ϵ for a linear SVM classifier), and a set of training features \mathcal{F} to represent the textual content of the reports. The final input parameters are the "bootstrap" threshold t , giving the minimal distance from the SVM hyperplane for a document to be included in the positive set on the next iteration, and the review count n giving the number of documents to be reviewed at the end of each iteration in the margin of the SVM hyperplane. The training set \mathcal{T} is comprised of four sets of documents: $\mathcal{T}.\mathcal{P}$, the real positives which have already been reviewed by the expert, $\mathcal{T}.\mathcal{N}$, the real negatives which have already been reviewed by the expert, $\mathcal{T}.p$, the positives automatically calculated by the previous model above the bootstrap threshold (initially provided by the expert in \mathcal{P}), and $\mathcal{T}.n$ a random sample of documents assumed to be negative, with a cardinality to balance the positive and negative examples. It is of course possible (and desirable) to give reviewed positives and negatives a higher weight than calculated positives/random neg-

<p>Input: corpus \mathcal{C}, initial positive set \mathcal{P}, training parameters, feature set \mathcal{F}, bootstrap threshold t, review count n</p> <p>Output: dimension vector \mathcal{D}_n</p> <pre> 1 $\mathcal{T} \leftarrow$ new training set; 2 $\mathcal{T}.p \leftarrow \mathcal{P}$; 3 $\mathcal{T}.\mathcal{P} \leftarrow \emptyset$; 4 $\mathcal{T}.\mathcal{N} \leftarrow \emptyset$; 5 $i \leftarrow 0$; 6 repeat 7 // Train model 8 $\mathcal{T}.n \leftarrow$ random sample with cardinality $(\mathcal{T}.\mathcal{P} + \mathcal{T}.p) - \mathcal{T}.\mathcal{N}$; 9 Train model \mathcal{M}_i using $\mathcal{T}.\mathcal{P} \cup \mathcal{T}.p$ as positive and $\mathcal{T}.\mathcal{N} \cup \mathcal{T}.n$ as negative examples and \mathcal{F} as features; 10 // Calculate dimension vector 11 $\mathcal{T}.p \leftarrow \emptyset$; 12 $\mathcal{D}_i \leftarrow$ new dimension vector; 13 foreach $doc \in \mathcal{C}$ indexed by j do 14 $\mathcal{D}_i[j] \leftarrow$ apply \mathcal{M}_i to doc_j using \mathcal{F}; 15 // Bootstrap calculated positives 16 if $\mathcal{D}_i[j] > t$ then add doc_j to $\mathcal{T}.p$; 17 end 18 // Review marginal documents closest to hyperplane 19 for n positive and n negative docs $\notin \mathcal{T}.\mathcal{P} \cup \mathcal{T}.\mathcal{N}$ closest to hyperplane do 20 Expert reviews doc_j; 21 Expert adds doc_j to $\mathcal{T}.\mathcal{P}$ or $\mathcal{T}.\mathcal{N}$; 22 end 23 $i \leftarrow i + 1$; 24 until expert satisfied; 25 return \mathcal{D}_i; </pre>

Algorithm 6.5.1: Iterative dimension training

atives. At each iteration, the system first trains a new model \mathcal{M}_i given the current training set \mathcal{T} . It then calculates a new dimension vector \mathcal{D}_i using the model \mathcal{M}_i . Within the algorithm, we'll assume \mathcal{D} contains a real positive or negative distance from the SVM hyperplane, although it is trivial to convert this to a yes/no answer by taking positives to be yes and negatives to be no. Finally, the system reconstructs \mathcal{T} as follows: $\mathcal{T}.p$ is automatically calculated by taking all documents where the distance from the hyperplane exceeds the bootstrap threshold. The expert is then asked to review the n documents closest to the hyperplane margin on both sides, and determine whether they are really positives or negatives, assigning them respectively to $\mathcal{T}.\mathcal{P}$ or $\mathcal{T}.\mathcal{N}$.

The assumption is that correctly reclassifying a small number of documents in these marginal areas allows us to converge much more quickly than a random review of documents.

The learning ends when the expert is satisfied with the dimension values assigned to documents—presumably when the hyperplane correctly distinguishes the majority of documents reviewed.

6.5.3 Simulation and results

In order to better understand the behaviour of the system and assess its usefulness, we ran several simulations using existing metadata as a validation criterion, substituting itself to the expert’s judgement. At each iteration, reclassifying the documents from the marginal areas is done based on whether they are *true* positives or negatives for the target metadata attribute.

We used the feature extractor described in section 4.1 (stems and stems n-grams) and a linear SVM classifier. For an estimation of the classifier’s margin we used the probabilities provided by the libLinear library, which are based on the distance between an item and the trained model’s hyperplane. The bootstrap threshold t is set at 0.8.

As a metric of performance, at each iteration we measured precision, recall and F1 scores for overlap between the documents identified by the system and the documents classified according to the target metadata attribute.

Table 6.6 shows the results of the simulation on a subset of the French DGAC corpus consisting of 44,191 documents. The task consists of creating a *Dimension* for bird strikes. The initial set $\mathcal{T}.p$ contains all documents that contain the word “oiseau”¹⁷. We have set the review count n to 10, meaning that at each iteration the 10 positive and 10 negative items with the lowest margins are submitted to the expert (or here, have their status revised according to their metadata).

The first row of table 6.6 shows the state of the system at query-time. The query has returned 1,534 documents. From those, 1,413 are considered *true* positives (have the occurrence category *BIRD*). The query is quite precise, with a precision of 92.11%, but its recall is 43.85%, meaning that less than half of the documents categorised as *BIRD* contain the word “oiseau” (in fact, most reports signal the exact species of bird encountered).

The second row shows the state of the system after a model has been trained on the initial set. $\mathcal{T}.p$ now contains the documents classified by the model. While no “human” reclassification has yet been performed, 346 new true positive documents have already been correctly identified by the system.

The subsequent rows show the state at each iteration. At iteration 3, for example the expert has reclassified a total of 40 documents (28 as positives and 12 as negatives). After the corresponding retraining, the system identifies

¹⁷“bird” in French.

176 more true positives as compared to the state at iteration 1. We can see that the F1 score is steadily increasing with each iteration, illustrating how expert input on a small amount of documents iteratively refines and tunes the classifier.

i	$\mathcal{T}.p$	$\mathcal{T}.P$	$\mathcal{T}.N$	True+	P (%)	R (%)	F1 (%)
0	1534	0	0	1413	92.11	43.85	59.42
1	1957	0	0	1759	89.88	54.59	67.93
2	2035	17	3	1804	88.65	55.99	68.63
3	2205	28	12	1935	87.76	60.06	71.31
4	2379	41	19	2104	88.44	65.30	75.13
10	3347	140	40	2877	85.96	89.29	87.59

Table 6.6: Results for *bird-strike* (DGAC corpus)

Table 6.7 shows the results of another simulation, this time on 7,025 documents from the ASRS database (selected on a temporal criterion from the corpus described in section 2.1.4). We simulated the search for incident reports where *confusion* was a factor and we use the *Human Factors* attribute of the *Person* entity as a validation criterion. We tested for those documents classified with the value *Confusion*. The initial query is the word “confusion”.

While this configuration is closer to the real-word use the system is intended for, it is also a much more difficult task than identifying bird-strikes. This difficulty can be estimated by training a simple classifier for this meta-data: our best configuration achieved only 66% F1-score, while we reach 95% for the *BIRD* category in the DGAC corpus.

Accordingly, the system performance is worse than in the previous scenario, but the behaviour is comparable. At iteration 3 the system has identified 253 more true positive documents with only 40 being submitted to the expert for validation. After 10 iterations, if the F1 score is still below 50%, recall has doubled.

i	$\mathcal{T}.p$	$\mathcal{T}.P$	$\mathcal{T}.N$	True+	P (%)	R (%)	F1 (%)
0	774	0	0	472	60.98	25.46	35.92
1	1048	0	0	574	54.77	30.96	39.56
2	1280	14	6	670	52.34	36.14	42.76
3	1443	24	16	725	50.24	39.10	43.98
4	1564	26	34	765	48.91	41.24	44.74
10	1936	57	123	900	46.49	48.54	47.49

Table 6.7: Results for *confusion* (ASRS corpus)

Globally these results are encouraging. They demonstrate that it is possible to better capitalise on the expert’s time and, with this type of active

iterative process, effectively “propagate” the judgement to a large proportion of the documents. While validating the general principle, these experiments also pose a number of questions. The most important one currently is the relationship between the initial query and the output of the system. We have observed that the system behaves differently depending on both the precision and the recall of the query. We also observed that, depending on the query, varying parameters such as the bootstrap threshold, the review count and the additional weight given to documents already reviewed have different effects and can greatly improve performance. As we can not have control on the query itself, we are searching for methods to automatically determine the optimal values of these parameters. We are also looking forward to building the graphical user interface and proposing the system to real word users. This will allow for much more realistic testing as we will be able to directly measure performance based on the proportion of yes/no judgements at each iteration.

6.5.4 Using *Dimensions* to produce KPIs: a use case

One very recent and interesting example came to us from a client of *Safety Data* who had once again “hijacked” the system and used it in a way we would not have easily imagined. The user trained a *Dimension* to recognise documents where fatigue was an issue. Rather than using the scores as a search criterion he used them in an aggregated form to track how “fatigue” distributes chronologically over the data and for each family of aircraft. The end¹⁸ result is a line chart, with one line per family, the sum of the score of the fatigue *Dimension* on the Y axis and time on the X axis. One could, for example, notice peaks of fatigue around charged periods (such as Christmas holiday season). This example shows how this particular method bridges the gap between the symbolic nature of documents and the need to address them quantitatively in order to study trends and look for patterns.

6.5.5 Possible application to similarity

There are two axes of reasoning when we think about how to benefit from such methods for a “better” document document similarity. The first one is do we really need document-document similarity when a system as the one we described addresses the same need as the *timePlot* system. The documents scoring high on a given *Dimension* are those that are most similar to the ones used for training. Besides, by encouraging the user to validate and invalidate some of the returned documents, he will quickly zero in on the exact aspect of similarity that interests him.

We can also leverage the output of such a system by considering that the model generated for a given “Dimension” is effectively capturing some of the

¹⁸Due to confidentiality concerns, we can not show the exact figures and go into more detail.

users knowledge. Thus, provided enough people use the system frequently we will have a large number of models, each dealing with a particular aspect of the data that was relevant for a user at a given time. Thus, for every document in the collection (and every new document available) we can compute a score for each “Dimension” These scores can then be compared by a normal cosine method to produce similarity scores based on aspects of interest. Such scores can be combined with any other vector-space model, such as topic modelling and would enrich the available facets that the system identifies. Besides, provided that this goes on in a live implementation, one could imagine that users go on and search for the latest “trends”, the ones that are by definition not captured in the coded data. Thus these facets will quickly turn up in the similarity scores.

6.6 Lessons learned and future work

In this chapter we looked at the notion of *similarity* from different perspectives. First, we asked the question of how to transform the concrete textual manifestations into a measure of relatedness. While using a direct mapping to a first order vector space model (such as the one in the *timePlot* system) is certainly a good start, directly mapping the terms to their corresponding dimensions is not always satisfactory. Multilingual databases are the most extreme example, but in a sense it can be viewed as just a radical manifestation of linguistic variation. So we used dimensionality reduction methods, such as ESA and *Topic Modelling* to produce an additional layer of transformation from the surface forms to a more abstract representation, closer to the meaning of the analysed documents. We verified the pertinence of the *topics* identified by topic modelling by comparing them to the metadata attributes of the ASRS corpus. This showed us that, while there is a general overlap, some of the *topics* not correlated to the metadata are very pertinent. On the other hand there are *topics* that are clearly noisy and uninterpretable. We believe that a similar situation will be observed for any dimensionality reduction technique.

On the other hand we looked at the notion of similarity in relation to the particular uses it is put to - identifying similar incidents for the purpose of helping experts find patterns in the data. We saw that users are willing to engage in an interaction with the system in order to influence the results, both to reduce noise, to achieve higher recall and to “isolate” a certain facet of the similarities identified by a system. Building on these observations, we leveraged the overlap between metadata and textual similarity and looked at methods that allow us to isolate facets of similarity and to “mask” the major dimensions in order to shed light on the secondary ones. In parallel, given the willingness of the users to interact with the system, we applied an active learning method, specifically designed for such an interaction.

Let us sum up the main advantages and drawbacks of the four methods presented in this chapter:

- **Filtering similarity** (§6.2) based on metadata attributes has the potential to isolate hidden secondary dimensions of relatedness. It follows that such an approach is dependant on the quality of the metadata and is applicable only in cases where the taxonomy's cover is sufficient. Also, having implemented it directly on surface forms, the method suffers from the same variation related issues as all first order representations (§3.1.2). Furthermore it is not clear in what manner a user might be involved in such an approach. Designing a usable interface which allows the filtering of similar results based on such criteria is no easy task. A potential extension to this line of reasoning, solving (some of) the variation related issues is incorporating domain knowledge in *Topic Modelling* or researching the interface between ESA and metadata.
- We applied the **second order similarity** (§6.3) method to multilingual databases with encouraging results. The second order vector space representation addresses language variability even at its most extreme form - texts written in different languages. The main problem with these methods is the quality of the resource used for the pivots. A term absent from the pivots, for example will not be accounted for in the final vector space, even if it is present in multiple documents. This kind of silence is particularly undesired in the context of safety related texts, where emergent phenomena are of a prime concern. The acronym designating a new system, for example will show up in many documents if the system is problematic. A potential solution to such a silence is to associate both *vectorisation* (Claveau, 2012) (using the indexed collection as pivots) and an external resource, but in such an approach one would lose the “conceptual” nature of the representation and should find means to control biases, potentially introduced by the uneven distribution of documents in the topical structure of the indexed collection.
- In **Topic Modelling** (§6.4) we see two distinct advantages. First, like ESA it reduces variation by producing more abstract representations. Moreover, (contrary to ESA) the topicality is abstracted in an endogenous manner, based on term cooccurrence within the indexed collection. This makes the technique suitable for collections where external resources are not available. It follows that the topical structure can be used as a basis for initial construction of taxonomies (an issue for many industries), by involving experts and interpreting the resulting topics. The main drawback of the approach we used was the need to determine in advance the number of topics and the flat nature of the resulting topical structure. These are nevertheless both issues with “vanilla” *Topic*

Modelling. The field is very active and different methods exist for automatically determining the number of topics (Arun et al., 2010), for producing hierarchical models (Teh et al., 2006) as well as for incorporating domain knowledge (Andrzejewski et al., 2009) and human judgement Hu et al. (2014). How these aspects articulate with one another is a different matter...

- The **Active Learning** (§6.5) approach was purposefully built around user involvement. We see a major potential in it and, as part of *Safety Data's* R&D we are continuing to develop it and searching for different ways to integrate it in a user-oriented process. Incorporating the *user's judgement* at every step of the process allows a higher level of control over the result, ultimately building a more accurate model. The *persistence* of the model is also an interesting factor. Once constructed it can be applied to newly introduced documents and (if needed) further refined by the users. We also see potential in using the method for collaborative approaches, where multiple users judge pertinence on the same model. One limitation is the opaqueness of the constructed model. It is difficult to explicitly know (and show) *why* a given score is attributed to a given document.

Of the four approaches we discussed, none is conclusive. We can not, at this point say that we have found *the* method to calculate similarity and built *the* application that will support pattern finding activities of safety experts. In one thing we are certain, however: That without simultaneously taking into consideration the **data**, the **domain** and the **user**, it is not possible to design such a system.

Conclusion

NLP as a component in a safety related information processing framework

In this thesis we explored the domain of aviation safety from a computational linguist's point of view with the goal to integrate NLP methods in the safety-related information processing framework of a high-risk industry. We focused on civil aviation, in part because we simply had access to this particular community and in part due to the fact that it is the industry where the collection and usage of occurrence data is the most advanced. This does not mean that the research presented here is not applicable to other industries. Every high-risk industry can use civil aviation as an example and many do (Barach and Small, 2000).

The main issue we addressed is efficient access to information, contained in large databases of incident and accident reports. The main point of entry to these collection today is still based on pre-established taxonomies, which are costly to maintain and do not easily scale. Natural language accounts, on the other hand, are both ubiquitous and do not pose a problem when data starts accumulating. We showed how, by exploiting them as input to Text Categorisation methods, it is possible to automatically improve the quality and the cover of the coded data. Furthermore we demonstrated that, even if natural language is varied and noisy, a sufficient portion of the variation can be accounted for. Therefore methods, such as textual similarity, based *only* on the natural language parts of incident and accident reports are sufficiently powerful to serve as basis for an application designed for browsing and searching collections of reports. Exploring the informational redundancy between taxonomies and natural language showed that robust bottom-up methods, such as *Topic Modelling* are capable of abstracting a topical structure close to the one described in the coded data. We also looked at possibilities to filter out the redundant information in order to capture the signal present only in the narrative parts.

How users relate to both the data and the application became clearer as we observed the actual uses, analysed feedback and conducted interviews with

safety experts. It became apparent that, combined with some specific expectations, such as high recall and transparent results, users were willing to engage in iterative and prolonged search strategies. Based on these observations, we developed an iterative process allowing example-based modelling of a given scenario, with the added benefit of producing a persistent model that can be applied to any document.

All in all, this thesis shows that text can be viewed not only as the vehicle of information between humans, but also as a resource that, when properly tapped and exploited, improves the overall quality of communication of safety-related information within a given system. We are confident that by incorporating NLP components in the information processing framework of a high-risk system, it is possible to conceive robust, bottom-up and scalable methods allowing more efficient use of large quantities of occurrence data, leading ultimately to an even better understanding of complex socio-technical systems and rendering them even more reliable in the future.

Furthermore, we are not alone in believing in the potential of language processing technologies applied to safety-related information. Today, *timePlot*'s successor, *PLUS*¹⁹, an industry-ready commercial application, built by a team of talented engineers, based largely upon the results of this research is a proof of the contributions NLP has to offer to the domain of risk management. The application was built with the same functionalities as the *timePlot* system and all the principles discussed in Chapter 5 also hold for it. At the time of this writing, the tool is operational or in the phase of being deployed at companies and government institutions, both in the civil aviation domain as well as in other sectors. Here is a (non exhaustive) list of *CFH - Safety Data*'s clients using or about to use the tool:

- **Civil Aviation:** *EASA, DGAC, Air France, Dassault Aviation, WFP*²⁰
- **Space:** *Astrium*
- **Rail:** *SNCF, RATP*
- **Energy:** *EDF*
- **Medical:** *UGE CAM*

As a consequence, we are also starting to notice how, by introducing custom tools and processes built around NLP technologies, practices within the community are starting to shift and the available data started to be looked upon in novel ways. While until recently report narratives were meant for “human eyes only”, now we start hearing voices from the community stating that automatic processing of languages can replace²¹ the current taxonomy-based paradigm.

¹⁹PLUS stands for Processing Language Upgrades Safety.

²⁰the World Food Program is in charge of the United Nation's transport operations.

²¹We personally find this claim a bit too extreme and overambitious, but we feel flattered nonetheless.

This thesis is coming to an end, but work on the subject is all but beginning. As we hope it has become clear in, we believe that access to data and to users are the essential prerequisites to successfully apply NLP to a given task. It also goes without saying that there are considerable engineering challenges associated with managing data and building applications.

We have reached now, with *CFH - Safety Data* the point where a comprehensive NLP-based solution for safety related data is becoming mature. It is therefore time for us to ask the question we were asking at the beginning of the thesis in a different manner. While our objective was then to apply NLP to risk management in civil aviation, now the question becomes: “what is the best way to proceed in the future?” In other words, how do we evaluate the contribution of the different NLP-components in this particular context, how do we choose between alternatives and how do we optimally parameter them?

Evaluation of NLP components in context

We worked in the particular context of risk management and we proposed NLP solutions to some of the needs we identified in the industry. Those solutions were bundled into applications and the applications put to the test in a real world environment. The most important positive evaluation is therefore the fact that people are using them and manage to accomplish their tasks in a more efficient manner than before they had access to the applications. Nevertheless when the system matures, evaluation of its performance becomes mandatory in order both to correctly parameter it and to be able to compare alternative approaches to any given problem and choose between them.

How to evaluate a given system (or component) is dependent on a variety of factors. Let us first introduce several basic concepts of the topic before presenting our vision on the subject.

According to Clark et al. (2013), “NLP is concerned with producing artifacts that accomplish tasks. The operative question in evaluating NLP is therefore the extent to which it produces the results for which it was designed” and gives the following dimensions to the topic:

- **Intrinsic vs. extrinsic evaluation:** Intrinsic criteria are those related to the system’s *objectives*, while extrinsic criteria are those related to the system’s *function*. (Jones and Galliers, 1995). In other words, performing intrinsic evaluation can be viewed as judging the system’s output according to a predefined criterion, say the proportion of correctly identified *tokens* by a *tokeniser* (§4.1.1). Performing extrinsic evaluation, on the other hand asks the question of how good the system is in the context of a specific task, treating the system as an “enabling technology” whose value resides in its contribution to a larger application (Clark et al., 2013). The tokeniser, being a part of a larger system

can be *extrinsically* evaluated by observing how changing its parameters improves the performance of a Information Retrieval system.

- **Component vs end-to-end evaluation:** As NLP systems are often modular, one might evaluate either the different components separately or evaluate the processing chain as a whole.
- **Automatic vs manual evaluations:** The third dichotomy is between automatic or manual evaluations. In an automatic evaluation, one simulates the behaviour, judgement and expectations of the user while in a manual evaluation, users are asked to directly judge the performance of the system. Automatic evaluations are easy to conduct (and can be run multiple times at no extra cost) but depend on accurate simulation of the users. Hand-crafting such simulations in the form of *gold standards* is labour intensive and, as Poibeau and Messiant (2008) discuss, can be an unnecessary burden for *intrinsic*, *component* evaluations in cases where a given system can be assessed *extrinsically* as a whole (*end-to-end*). Manual evaluations, on the other hand, are costly to conduct, slow and can generally be run only a few times. They also suffer from a number of biases such as the inherent inconsistency of humans to judge what is “good”.

Given that the objective of this thesis is to equip specific users with tools that satisfy specific needs, we look at the question of evaluating the system(s) from an *extrinsic* perspective and on an *end-to-end* basis. Given the variety of tasks, we came to think about the *possibility* to conduct such evaluations. We see it as a spectrum, ranging from the necessary and straightforward approach to cases where conducting formal evaluations is unreasonably difficult:

1. **Necessary and straightforward:** This is the case of automatic text categorisation, a case where *extrinsic* and *intrinsic* evaluation overlap and where measuring performance is an integral part of the supervised classification task. The system’s task is to assign categories to documents. We can directly measure the system’s performance and compare it to documents classified by humans. The system we present in Section 3.2.3 was evaluated in this manner, and the results directed the particular choice of industrialisation.
2. **Feasible and straightforward:** This is the case where an existing protocol can be adapted and tweaked in order to reflect a particular desired outcome. Such is the case with the multilingual second order similarity experiments (§6.2). We performed an automatic evaluation by using a parallel corpus and measuring the system’s capabilities of finding translations among the reports. While this task proved us that the system worked, it is still an artificially constructed task. It does not

inform us on the performance of the systems when presented the data it is intended for (Multilingual databases of incident reports (§2.1.5)).

3. **Feasible but costly:** For IR systems, for example, the evaluation needs to simulate the users expectations. The TREC competitions provide ample examples of both the difficulty of the task and the necessity for a great number of separate simulations in order to cover different contexts of IR searching in a firm’s internal document collection (Balog et al., 2008) is different than searching the web (Collins-Thompson et al., 2014). Building a similar evaluation protocol for incident and accident data is certainly feasible but will be a very costly enterprise. For the time being it is not reasonable, at least until we have both sufficiently refined our understanding of the information needs and sufficiently observed real world user interactions with an IR system in this particular context.
4. **Unreasonably costly:** Performing extrinsic end-to-end evaluations on a system identifying similar documents or on an application that aids experts in identifying patterns raises a lot of questions. In both cases the task is too vague in order to be formally defined. In the case of *similarity*, for example, we have an intuitive understanding of the notion of similarity, but in order to conduct an evaluation of such a system, we would need to construct a model taking into account *all* the facets and dimensions of similarity, to which to compare the system’s output. In the case of a *pattern-finding* activity, the task is so abstract that a formal definition of “a pattern” and how it relates to the data and the context is practically unfeasible. In consequence, one way we imagine an evaluation of such a system is in the form of a manual and simulation-based approach. A sort of “treasure-hunt”, where a domain expert introduces a set of related documents into a collection of unrelated reports and another expert tries to identify them using the system. While imaginable, in order to be valid and significant one should control parameters such as the unrelated nature of the collection, the validity of the artificially introduced patterns (hence the need for a peer-based approach) and the fact that the users are unbiased. One would also have to repeat the exercise many times with different users and scenarios. These variables render such an approach to all extents unreasonably difficult. Should this impossibility at evaluating such system stop us from attempting to build one?

We are certain that the bulk of the tasks NLP applications in the domain of risk management will fall in the last category. Which brings us to reconsider the current evaluation paradigms, apply them whenever possible, but also look for complementary means of ensuring acceptable performance. Our approach is twofold and revolves around the notions of *transparency* and user *involvement*.

- Incorporating **transparency** into the system means ensuring that the underlying processing remain both comprehensible and visible to the end user. It became clear from the *timePlot* system that even basic visual cues such as highlighting shared terms gives the user information about *why* a given document was returned. In other words to understand the process that leads to this particular result. He can then effortlessly assess the pertinence of the results and, if needed, interact with the system to either filter the results or modify his initial query. An example of incorporated transparency is the highlighting of similar terms in the *timePlot* system (§5.2).
- User **involvement** is transparency viewed backwards. Explaining to the users *what* the system does and *how* it does it, makes them more able to perform their tasks. Of course, this partly depends on user base, but in our case we were very lucky²². People dealing with safety issues know that machines are imperfect and demand to understand both the capabilities and the limits of the systems they are interacting with, regardless if it is a helicopter or a search engine. We largely based the development of the active learning approach (§6.5) around the notion of user involvement.

We would also to be able to act on the smallest possible level of grain and thus prefer *modular* approaches to monolithic approaches for any given task. In other words we would prefer having the possibility to intervene at a very²³ small scale in order to adjust a particular problem or error produced by the system.

It follows that, by building *transparent* and *modular* processing, while maintaining a channel for feedback from an informed (and *involved*) user will allow us to act *a posteriori* and act based on a concrete example of an undesirable result.

Regarding the work we presented in this thesis, the main consequence of these considerations is that they make us reconsider using (opaque) dimensionality reduction techniques as the **unique abstraction layer** between text and representation. As a consequence, we will start looking for ways to progressively integrate symbolic and knowledge rich methods in the tools we propose.

²²Such a position would be completely unthinkable and counterproductive for other types of applications, such as web search engines, where people constantly try to “game” the system for higher visibility in the search ranks. As a consequence the system’s performance depends on the opaqueness of the underlying processing.

²³A trivial example is the one we mentioned about the stemming of the word “laser” to “las” (§3.1.2) in this case we do not want to have to reinvent a stemming algorithm, but to be able to manually add an exception.

Towards symbolic knowledge rich methods

Whether it is a Text Classification component, a Search Engine or a system for identifying similar documents, the ultimate goal of the NLP component is to account for the *meaning* of the input text, manipulate it, compare it either to a model, to the *meaning* of a query or the *meaning* of another text. The main obstacle to overcome before one can even start talking about meaning (as opposed to simple lexical overlap), is that of variation (§3.1.2, §4.1.2). In other words, to move further away from noisy surface forms and more toward abstract unambiguous meaning representations.

There are basically two ways to address the problem. One could either *describe* meaning and its relation to language or one could (try to) *abstract* it statistically, based on a (very large) collection of texts and on frequency and patterns of cooccurrence.

In light of the manifest difficulty of extrinsically evaluating NLP components in context of risk management, we have to look back critically on the dimensionality reduction techniques we used in this thesis. Methods such as ESA (§6.2) or *Topic Modelling* (§6.4) produce such more abstract representations. Whether “conceptual dimensions” or “topics” these methods map the surface forms to vectors of a higher order dimensions that account for some aspect of the meaning of the text they encounter. While they can, and are being applied with relative success to a number of tasks, this family of methods present a number of inherent limits. These are:

- They tend to be opaque. The model (automatically) constructed by any such techniques is at best partially interpretable.
- It follows that they offer little if no direct control over the mappings. It is difficult²⁴ to act on a small scale and, say, add or remove variants of a given term.
- They are monolithic. Because they account for different aspects of how language varies, they tend to be applied as single components with basic tokens as input and “conceptual” dimensions as an output.

As a function of these three limitations, such statistical methods, relying on quantity and frequency all reach a ceiling at some point. And when the ceiling is reached, there is not much one can do than “stir the probabilistic cauldron”, maybe add more data to the system or change for a better and newer technique hoping to improve performance. This entails that using such methods implies designing formal evaluation criteria. This, as we saw in the previous section is, in some cases impossible or at least very costly. In this light

²⁴Work is however being done to incorporate interactivity in Topic Modelling (Hu et al., 2014).

we will continue to test and use dimensionality reduction in those areas where either an objective formal evaluation is possible (such as Text Categorisation) or where they come as an aid to humans for necessary modelling tasks, such as building and maintaining lexicons.

The other option as we saw is to *describe* meaning and its relationship to its primary vehicle (text). A knowledge-rich approach implies constructing a world model as a central resource for extracting meaning from text. For Nirenburg (2004) a knowledge rich approach for NLP is comprised of:

1. A set of static knowledge sources: An ontology, a fact-repository and a lexicon (mapping the ontology to natural language).
2. Knowledge representation languages for specifying meaning structures.
3. A set of processing modules - semantic analysers.

And while to Turney and Pantel (2010) it “seems possible that all of the semantics of human language might one day be captured in some kind of Vector Space Model”, we know that the semantics of language can be *described* and, provided there is a resource with sufficient cover and scale, can be applied to extract the underlying meanings from text in an understandable form.

The main criticism of knowledge-rich methods is that the sheer effort needed to construct the model is forbidding their practical application outside small and controlled lab environments. In other words they lack robustness. Such approaches were mainstream from the 1960’s to the 1990’s but were gradually replaced by statistical NLP in the late 1990’s (Spärck-Jones, 2001). Now they are starting to (spectacularly) come back with systems such as *IBM’s Watson* (Gliozzo et al., 2013) or *Inquire* (Chaudhri et al., 2013).

Lannoy (1996) discusses a potential application of full scale semantic representations to the domain of risk management only to show that the complexity of the modelling effort is forbidding. It also illustrates two fundamental problems with a number of such approaches:

- First the goal of this particular “expert system” is to, in a nutshell, extract meaning from the texts of incident reports then reason on the meaning representations in order to identify the causes of an incident. In other words replace the expert by an automatic process. While we can not but credit such a goal for its ambition, we believe that it completely misses the point, mainly that human experts need not be *replaced* but *assisted* in working with the ever increasing amounts of data and the information it contains.
- Secondly, (given the ambitious goal) the success of such a system depends upon the completeness of *all* the modelling (all levels of language and a complete model of the domain) in its entirety before it is capable of delivering any usable results. In practice knowledge rich methods

need not be extremely sophisticated and a result of a year long modelling effort in order to deliver a practical improvement of an NLP component. Simply listing all the species of birds and attaching them to the *concept* of BIRD will give higher recall for an IR system when searching for bird-strikes.

We nevertheless are confident that, in the particular context of risk management in civil aviation, knowledge-rich methods are both inevitable if we want to achieve long term progress and feasible as a basis for robust NLP applications. Here are the main reasons:

- **Existing modelling culture:** In civil aviation (and risk management) there is a long lasting modelling tradition as the various accident models (Qureshi, 2007), such as (parts of) ADREP (§2.2.3.5), the Bow-tie model (§2.1.5) or activity models describing the sequence of a commercial flight (Ale et al., 2005) testify. moreover some of these models are instantiated over large numbers of accident and incident reports.
- **Large quantities of natural language data:** Accident reports in particular are ideal text mining candidates for knowledge extraction (Toussaint, 2011; Poibeau, 2003) and can also serve as a basis for partially populating existing resources.
- **Expert availability:** Safety experts we have worked with both are in high demand of means to better access information and understand the challenges that such systems pose. They are willing to participate in a modelling effort, provided that it is conceived in such a way as to have short to medium-term returns in the form of better and more usable applications.
- **Scale:** Civil aviation is a global system. Accounting for a small subset of domain specific phenomena (such as pilot fatigue for example) will be of interest for a very large community.
- **High stakes:** Aviation accidents are incredibly costly and even a slight improvement of safety saves a lot of money.

Adding that, today contrary to the 90's:

- The technology to build systems that scale to terabytes of data exists.
- Users generate more and more quality information and are being exploited for knowledge acquisition (Lafourcade, 2007; Wang et al., 2012).
- Robust large scale methods for knowledge extraction from text are available, partially as an answer to the availability of large quantities of annotated texts (Bellot et al., 2014).

- The technology, infrastructure and methodologies to build truly global web applications is mature.

We believe that in the future knowledge-rich methods will be the basis of the NLP-components civil aviation's information processing framework.

Bibliography

- Ahsan, S., Alshomrani, S., and Hassan, A. (2013). Semantic data mining for security informatics: Opportunities and challenges. *Life Science Journal*, 10(12s).
- Ale, B. J. M., Bellamy, L. J., Roelen, A. L. C., Cooke, R. M., Goossens, L. H. J., Hale, A. R., Kurowicka, D., and Smith, E. (2005). Development of a causal model for air transport safety. page 107–116.
- Allan, J. (2006). A heuristic risk assessment technique for birdstrike management at airports. *Risk Analysis*, 26(3):723–729.
- Amalberti, R. (2001). The paradoxes of almost totally safe transportation systems. *Safety Science*, 37(2–3):109–126.
- Andrzejewski, D., Zhu, X., and Craven, M. (2009). Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 25–32. ACM.
- Arampatzis, A. T., Van Der Weide, T. P., van Bommel, P., and Koster, C. H. (2000). Linguistically motivated information retrieval. *Encyclopedia of Library and Information Science: Volume 69-Supplement 32*, page 201.
- Arun, R., Suresh, V., Madhavan, C. V., and Murthy, M. N. (2010). On finding the natural number of topics with latent dirichlet allocation: Some observations. In *Advances in Knowledge Discovery and Data Mining*, pages 391–402. Springer.
- ASRS (2014). ASRS coding taxonomy.
- Balog, K., Thomas, P., Craswell, N., Soboroff, I., Bailey, P., and De Vries, A. P. (2008). Overview of the TREC 2008 enterprise track. Technical report, DTIC Document.
- Barach, P. and Small, S. D. (2000). Reporting and preventing medical mishaps: lessons from non-medical near miss reporting systems. *BMJ*, 320(7237):759–763.

- Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.
- Bellot, P., Bonnefoy, L., Bouvier, V., Duvert, F., and Kim, Y.-M. (2014). Large scale text mining approaches for information retrieval and extraction. In Faucher, C. and Jain, L. C., editors, *Innovations in Intelligent Machines-4*, volume 514 of *Studies in Computational Intelligence*, pages 3–45. Springer International Publishing.
- Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.
- Biber, D. (1993). Representativeness in corpus design. *Literary and linguistic computing*, 8(4):243–257.
- Bird, F. E. (1984). *Management Guide to Loss Control*. NOSA Safety Centre.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.
- Blaser, S., Agnew, S., Kannan, N., and Ng, P. (2004). High volume targeting of advertisements to user of online service. US Patent 6,757,661.
- Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.
- Bourigault, D. (1993). Analyse syntaxique locale pour le repérage de termes complexes dans un texte. *TAL. Traitement automatique des langues*, 34(2):105–117.
- Campello Rodrigues, J. P. (2013). *Classification automatique de séquences textuelles de rapports d’accidents aériens : une approche rhétorique*. Mémoire de M1, Université de Toulouse 2 - Le Mirail, Toulouse, France.
- Chang, J., Gerrish, S., Wang, C., Boyd-graber, J. L., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc.
- Chaudhri, V. K., Cheng, B., Overholtzer, A., Roschelle, J., Spaulding, A., Clark, P., Greaves, M., and Gunning, D. (2013). Inquire biology: A textbook that answers questions. *AI Magazine*, 34(3):55–72.

- Chevalier, M. (2011). *Usagers & Recherche d'Information*. Habilitation à diriger des recherches, Université Paul Sabatier, Toulouse, France.
- Clark, A., Fox, C., and Lappin, S. (2013). *The Handbook of computational linguistics and natural language processing*. Willey-Blackwell.
- Claveau, V. (2012). Vectorisation, okapi et calcul de similarité pour le TAL : pour oublier enfin le TF-IDF. In *Actes de TALN*, page 85–98, Grenoble.
- Collins-Thompson, K., Bennett, P., Diaz, F., Clarke, C. L., and Voorhees, E. M. (2014). TREC 2013 web track overview. *Ann Arbor University, Tech. Rep.*
- Cormack, G. V. (2007). Email spam filtering: A systematic review. *Foundations and Trends in Information Retrieval*, 1(4):335–455.
- Croft, W. B., Metzler, D., and Strohman, T. (2010). *Search engines: Information retrieval in practice*. Addison-Wesley Reading.
- Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- DGAC (2013). Rapport sur la sécurité aérienne 2013. Technical report, DGAC.
- Feldman, R. and Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press.
- Fellbaum, C. (1998). *WordNet*. Wiley Online Library.
- Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI*, page 1606–1611, Hyderabad, India.
- Glozzo, A., Biran, O., Patwardhan, S., and McKeown, K. (2013). Semantic Technologies in IBM Watson. In *Proceedings of the Fourth Workshop on Teaching NLP and CL*, pages 85–92, Sofia, Bulgaria.
- Gonzalo, J., Verdejo, F., Chugur, I., and Cigarran, J. (1998). Indexing with wordnet synsets can improve text retrieval. In *Proceedings of COLING/ACL'98 Workshop Usage of WordNet for NLP*.
- Grefenstette, G. and Tapanainen, P. (1994). *What is a word, what is a sentence?: problems of Tokenisation*. Rank Xerox Research Centre.
- Heinrich, H., Petersen, D., Roos, N., Brown, J., and Hazlett, S. (1980). *Industrial Accident Prevention: A Safety Management Approach*. McGraw-Hill.

- Hermann, E., Leblois, S., Mazeau, M., Bourigault, D., Fabre, C., Travadel, S., Durgeat, P., and Nouvel, D. (2008). Outils de traitement automatique des langues appliqués aux comptes rendus d’incidents et d’accidents. In *16e Congrès de Maîtrise des Risques et de Sécurité de Fonctionnement, Avignon*.
- Hernandez, N. (2005). Ontologies de domaine pour la modélisation du contexte en recherche d’information. *Thèse de doctorat, Institut de Recherche en Informatique de Toulouse*.
- Ho, C.-H. and Lin, C.-J. (2012). Large-scale linear support vector regression. *Journal of Machine Learning Research*, 13:3323–3348.
- Hofmann, T. (1999a). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc.
- Hofmann, T. (1999b). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM.
- Hollnagel, E. (2004). *Barriers and Accident Prevention*. Ashgate Publishing, Ltd.
- Hu, Y., Boyd-Graber, J., Satinoff, B., and Smith, A. (2014). Interactive topic modeling. *Machine learning*, 95(3):423–469.
- ICAO (2001). International civil aviation convention - annex 13 - incident reporting, data systems and information exchange.
- ICAO (2014). ICAO 2014 safety report. Technical report, ICAO.
- Jansen, B. J., Booth, D. L., and Spink, A. (2009). Patterns of query reformulation during Web searching. *Journal of the American Society for Information Science and Technology*, 60(7):1358–1371.
- Johnson, C. W. (2003). *Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting*. University of Glasgow Press, Glasgow, Scotland. ISBN 0-85261-784-4, available in in electronic form.
- Jones, K. S. and Galliers, J. R. (1995). *Evaluating Natural Language Processing Systems: An Analysis and Review*. Springer Science Business Media.
- Kilgariff, A. and Tugwell, D. (2001). Word sketch: Extraction and display of significant collocations for lexicography. In *Proceedings of ACL 2001*, pages 32–38.

- Kim, S.-M. and Hovy, E. (2006). Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 1–8. Association for Computational Linguistics.
- Kiss, T. and Strunk, J. (2006). Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.
- Kristjansson, T., Culotta, A., Viola, P., and Callum, A. M. (2004). Interactive information extraction with constrained conditional random fields. In *proceeding of the Conference of the American Association for Artificial Intelligence (AAAI)*, San Jose, CA.
- Kronrod, A. and Engel, O. (2001). Accessibility theory and referring expressions in newspaper headlines. *Journal of Pragmatics*, 33(5):683–699.
- Kules, W., Wilson, M. L., Shneiderman, B., et al. (2008). From keyword search to exploration: How result visualization aids discovery on the web.
- Lafourcade, M. (2007). Making people play for lexical acquisition with the jeuxdemots prototype. In *Proceedings of SNLP’07: 7th international symposium on natural language processing*, page 7.
- Lannoy, A. (1996). Analyse utomatique de texte libre. application au codage et à la validation de fiches de retour d’expérience. In *Actes de $\lambda\mu$* .
- Lukashenko, R., Graudina, V., and Grundspenkis, J. (2007). Computer-based plagiarism detection methods and tools: an overview. In *Proceedings of the 2007 international conference on Computer systems and technologies*, page 40. ACM.
- Macrae, C. (2007). Analyzing Near-Miss events: Risk management in incident reporting and investigation systems. *Centre for Analysis of Risk and Regulation*.
- Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.
- Manuele, F. (2013). Reviewing heinrich: Dislodging two myths from the practice of safety. In *On the Practice of Safety*, pages 234–256. John Wiley & Sons, Inc.

- Marchionini, G. and White, R. (2007). Find what you need, understand what you find. *International Journal of Human - Computer Interaction*, 23(3):205–237.
- Maron, M. E. (1961). Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417.
- Menzel, R. (2004). ICAO safety database strengthened by introduction of new software. *ICAO Journal*, 59(3):19.
- Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Morlane-Hondère, F. (2013). *Evaluation of resources provided by automatic distributional analysis : a linguistic approach*. Phd thesis, Université Toulouse le Mirail - Toulouse II.
- NASA (2014). ASRS program briefing. Technical report, NASA.
- Nirenburg, S. (2004). *Ontological semantics*. Language, speech, and communication. MIT Press.
- NTSB (1957). Midair collision, accident investigation report, trans world airlines lockheed 1049a n6902c and united air lines douglas DC-7 n6324c, over the grand canyon, arizona, june 30, 1956. Aircraft Accident Report 1-0090, National Transportation Safety Board.
- NTSB (1975). Trans world airlines, inc. boeing 727-231, n54328 berryville, virg december 1, 1974. Aircraft Accident Report NTSB - AAR -7516, National Transportation Safety Board.
- Olsson, F. (2009). A literature survey of active machine learning in the context of natural language processing. Technical report, Swedish Institute of Computer Science.
- Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.
- Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 1320–1326.
- Pelletier, F. J. (1994). The principle of semantic compositionality. *Topoi*, 13(1):11–24.
- Poibeau, T. (2003). Extraction automatique d’information (du texte brut au web sémantique).

- Poibeau, T. and Messiant, C. (2008). Do we still Need Gold Standards for Evaluation? In *Language Resource and Evaluation Conference*, pages –, Morocco.
- Power, R., Scott, D., and Bouayad-Agha, N. (2003). Document structure. *Computational Linguistics*, 29(2):211–260.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.
- Qureshi, Z. H. (2007). A review of accident modelling approaches for complex socio-technical systems. In *Proceedings of the Twelfth Australian Workshop on Safety Critical Systems and Software and Safety-related Programmable Systems - Volume 86*, SCS '07, page 47–59. Australian Computer Society, Inc.
- RAE (1954). Report on comet accident investigation. Technical report, Royal Aircraft Establishment (Great Britain) and Great Britain. Supply Ministry.
- Rasmussen, J. (1997). Risk management in a dynamic society: a modelling problem. *Safety science*, 27(2-3):183–213.
- Rasmussen, J., Svedung, R. ., and Svedung, I. (2000). *Proactive Risk Management in a Dynamic Society*. Swedish Rescue Services Agency.
- Reason, J. (2000). Human error: models and management. *BMJ*, 320(7237):768–770.
- Rebeyrolle, J., Jacques, M.-P., and Péry-Woodley, M.-P. (2009). Titres et intertitres dans l’organisation du discours. *Journal of French Language Studies*, 19(02):269–290.
- Řehůřek, R. and Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Malta.
- Ribeiro, N. (2014). Visualisation interactive de similarité textuelle - intervention du topic modeling. Master’s thesis, Université de Toulouse.
- Robertson, S. E. (1977). Theories and models in information retrieval. *Journal of Documentation*, 33(2):126–148.
- Roese, N. J. and Vohs, K. D. (2012). Hindsight bias. *Perspectives on Psychological Science*, 7(5):411–426.
- Roth, D. and Small, K. (2006). Margin-based active learning for structured output spaces. In *Proceedings of the European Conference on Machine Learning (ECML)*, page 413–424.

- Sahlgren, M. (2005). An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE*, volume 5.
- Salton, G. and McGill, M. J. (1986). *Introduction to modern information retrieval*. McGraw-Hill, Inc.
- Salton, G., Wong, A., and Yang, C. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.
- Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*, volume 12, pages 44–49.
- Shannon, C. E. (1948). A mathematical theory of communication. 27(3):379–423.
- Singh, A., Rose, C., Visweswariah, K., Chenthamarakshan, V., and Kambhatla, N. (2010). Prospect: a system for screening candidates for recruitment. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 659–668. ACM.
- Sorg, P. and Cimiano, P. (2012). Exploiting wikipedia for cross-lingual and multilingual information retrieval. 74(0):26 – 45.
- Spärck-Jones, K. (2001). Natural language processing: a historical review.
- Spärck-Jones, K. (2004). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 60(5):493–502.
- Spärck Jones, K., Walker, S., and Robertson, S. (2000). A probabilistic model of information retrieval: development and comparative experiments. 36(6):779–808.
- Stephens, C., Ferrante, O., Olsen, K., and Sood, V. (2008). Standardizing international taxonomies.
- Tanguy, L., Tulechki, N., Urieli, A., Hermann, E., and Raynal, C. (2015). Natural language processing for aviation safety reports: from classification to interactive analysis. *Computers in Industry*. In print.
- Tanguy, L., Urieli, A., Calderone, B., Hathout, N., and Sajous, F. (2011). A multitude of linguistically-rich features for authorship attribution. In *PAN Lab at CLEF*.
- Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476).

- Teufel, S. and Moens, M. (2002). Summarizing scientific articles – experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.
- Thibert, C. (2014). *Marqueurs de la Fatigue et du Stress dans les Rapports d’Incidents Aériens*. Memoire de M1, Université de Toulouse 2 - Le Mirail, Toulouse, France.
- Toussaint, Y. (2011). *Text Mining: Symbolic methods to build ontologies and to semantically annotate texts*. Habilitation à diriger des recherches, Université Henri Poincaré - Nancy I.
- Tulechki, N. and Tanguy, L. (2012). Effacement de dimensions de similarité textuelle pour l’exploration de collections de rapports d’incidents aéronautiques. In *Conférence annuelle du Traitement Automatique des Langues Naturelles (TALN)*, Grenoble, France.
- Tulechki, N. and Tanguy, L. (2013). Similarité de second ordre pour l’exploration de bases textuelles multilingues. In *20e conférence du Traitement Automatique du Langage Naturel (TALN)*, Sables d’Olonne, France.
- Turner, B. (1992). *Sociology of safty*, page 186–201. McGaw Hill.
- Turney, P. D. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In Raedt, L. D. and Flach, P., editors, *Machine Learning: ECML 2001*, number 2167 in Lecture Notes in Computer Science, pages 491–502. Springer Berlin Heidelberg.
- Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. 37:141–188.
- Urieli, A. (2013). *Robust French syntax analysis: reconciling statistical methods and linguistic knowledge in the Talismane toolkit*. PhD thesis, Université de Toulouse II le Mirail.
- Vapnik, V. N. (2006). *Estimation of dependences based on empirical data ; Empirical inference science: afterword of 2006*. Information science and statistics. Springer, New York, N.Y.
- Vaughan, D. (1996). *The Challenger Launch Decision: Risky Technology, Culture and Deviance at NASA*. University of Chicago Press, Chicago.
- Véronis, J. (2000). *Parallel Text Processing: Alignment and use of translation corpora*. MIT Press.
- Wang, J., Kraska, T., Franklin, M. J., and Feng, J. (2012). Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494.

- Winder, C. and Michaelis, S. (2005). Aircraft air quality malfunction incidents: causation, regulatory, reporting and rates. In *Air Quality in Airplane Cabins and Similar Enclosed Spaces*, pages 211–228. Springer.